



# **NAVAL POSTGRADUATE SCHOOL**

**MONTEREY, CALIFORNIA**

## **THESIS**

**TEMPERATURE RISE INDUCED BY A  
ROTATING/DITHERING LASER BEAM ON A FINITE  
SOLID**

by

Tsuwei Tan

December 2010

Thesis Co-Advisors:

Hong Zhou  
Gamani Karunasiri

**Approved for public release; distribution is unlimited**

THIS PAGE INTENTIONALLY LEFT BLANK

<b>REPORT DOCUMENTATION PAGE</b>			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
<b>1. AGENCY USE ONLY (Leave blank)</b>		<b>2. REPORT DATE</b> December 2010	<b>3. REPORT TYPE AND DATES COVERED</b> Master's Thesis	
<b>4. TITLE AND SUBTITLE</b> Temperature Rise Induced by a Rotating/Dithering Laser Beam on a Finite Solid			<b>5. FUNDING NUMBERS</b>	
<b>6. AUTHOR(S)</b> Tsuwei Tan				
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> Naval Postgraduate School Monterey, CA 93943-5000			<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>	
<b>9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> N/A			<b>10. SPONSORING/MONITORING AGENCY REPORT NUMBER</b>	
<b>11. SUPPLEMENTARY NOTES</b> The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number _____.				
<b>12a. DISTRIBUTION / AVAILABILITY STATEMENT</b> Approved for public release; distribution is unlimited			<b>12b. DISTRIBUTION CODE</b>	
<b>13. ABSTRACT (maximum 200 words)</b>  <p>High energy laser weapons have been evolving progressively in recent years. These weapons deliver high-intensity beams to a target and can instantly destroy or burn it. They may cause potential threats to Navy ships, computer networks, guided missiles, and satellites in orbit. In order to reduce our military's vulnerability to high energy laser weapons, one possible countermeasure is to rotate or rock the object itself when it is hit by the laser beam.</p> <p>The main purpose of this thesis is to investigate the relationship between the speed of a rotating/dithering laser beam and the maximum temperature rise induced by the laser beam on a finite solid. We have investigated extensively the numerical solutions for the transient temperature rise in both one-dimensional (1-D) and two-dimensional (2-D) finite solids due to rotating/dithering laser beams. Our mathematical approaches include the eigenfunction expansion method, the Crank-Nicolson method, the Fast Fourier Transform method, and COMSOL for 1-D and 2-D cases. We have employed COMSOL to solve the 3-D nonhomogeneous heat equation.</p> <p>This thesis provides the first study that we know of on the effect of rotating/dithering laser beams on a finite target. Our results are consistent with previous analytical studies on semi-infinite regions. The quantitative relationship between maximum temperature rise and laser beam rotating speed, which is presented in this thesis, can be used as a general guide for adjusting the speed of rotation of the target in order to prevent the maximum temperature rise from reaching the melting point of the target.</p>				
<b>14. SUBJECT TERMS</b> Separations of variables, eigenfunction expansion, Crank-Nicolson method, Fast Fourier Transform method, nonhomogeneous heat equation, laser beams.			<b>15. NUMBER OF PAGES</b> 117	
			<b>16. PRICE CODE</b>	
<b>17. SECURITY CLASSIFICATION OF REPORT</b> Unclassified	<b>18. SECURITY CLASSIFICATION OF THIS PAGE</b> Unclassified	<b>19. SECURITY CLASSIFICATION OF ABSTRACT</b> Unclassified	<b>20. LIMITATION OF ABSTRACT</b> UU	

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public release; distribution is unlimited**

**TEMPERATURE RISE INDUCED BY A ROTATING/DITHERING LASER  
BEAM ON A FINITE SOLID**

Tsuwei Tan

Lieutenant, Republic of China Navy (Taiwanese Navy),  
B.S., United States Naval Academy at Annapolis, 2004

Submitted in partial fulfillment of the  
requirements for the degrees of

**MASTER OF SCIENCE IN APPLIED PHYSICS**

and

**MASTER OF SCIENCE IN APPLIED MATHEMATICS**

from the

**NAVAL POSTGRADUATE SCHOOL  
December 2010**

Author: Tsuwei Tan

Approved by: Hong Zhou  
Thesis Co-Advisor

Gamani Karunasiri  
Thesis Co-Advisor

Carlos Borges  
Chairman, Department of Applied Mathematics

Andres Larraza  
Chairman, Department of Physics

THIS PAGE INTENTIONALLY LEFT BLANK

## ABSTRACT

High energy laser weapons have been evolving progressively in recent years. These weapons deliver high-intensity beams to a target and can instantly destroy or burn it. They may cause potential threats to Navy ships, computer networks, guided missiles, and satellites in orbit. In order to reduce our military's vulnerability to high energy laser weapons, one possible countermeasure is to rotate or rock the object itself when it is hit by the laser beam.

The main purpose of this thesis is to investigate the relationship between the speed of a rotating/dithering laser beam and the maximum temperature rise induced by the laser beam on a finite solid. We have investigated extensively the numerical solutions for the transient temperature rise in both one-dimensional (1-D) and two-dimensional (2-D) finite solids due to rotating/dithering laser beams. Our mathematical approaches include the eigenfunction expansion method, the Crank-Nicolson method, the Fast Fourier Transform method, and COMSOL for 1-D and 2-D cases. We have employed COMSOL to solve the 3-D nonhomogeneous heat equation.

This thesis provides the first study that we know of on the effect of rotating/dithering laser beams on a finite target. Our results are consistent with previous analytical studies on semi-infinite regions. The quantitative relationship between maximum temperature rise and laser beam rotating speed, which is presented in this thesis, can be used as a general guide for adjusting the speed of rotation of the target in order to prevent the maximum temperature rise from reaching the melting point of the target.

THIS PAGE INTENTIONALLY LEFT BLANK



## TABLE OF CONTENTS

<b>I.</b>	<b>INTRODUCTION.....</b>	<b>1</b>
<b>II.</b>	<b>ANALYTICAL SOLUTION FOR A TRANSIENT TEMPERATURE DISTRIBUTION DUE TO A ROTATING OR DITHERING LASER BEAM IN A FINITE SOLID: EIGEN-FUNCTION EXPANSION METHOD .....</b>	<b>3</b>
<b>III.</b>	<b>NUMERICAL SOLUTION FOR A TRANSIENT, ONE-DIMENSIONAL TEMPERATURE DISTRIBUTION IN A FINITE ROD DUE TO A DITHERING LASER BEAM.....</b>	<b>15</b>
<b>A.</b>	<b>THE CRANK-NICOLSON METHOD .....</b>	<b>15</b>
<b>B.</b>	<b>THE FAST FOURIER TRANSFORM (FFT) METHOD .....</b>	<b>24</b>
<b>C.</b>	<b>COMSOL MULTIPHYSICS 4.0A .....</b>	<b>29</b>
<b>D.</b>	<b>COMPARISON ON A MODEL PROBLEM .....</b>	<b>30</b>
<b>E.</b>	<b>REAL PROBLEM SIMULATION (STEEL AISI 4340) .....</b>	<b>30</b>
<b>IV.</b>	<b>NUMERICAL SOLUTION FOR A TRANSIENT, TWO-DIMENSIONAL TEMPERATURE DISTRIBUTION IN A FINITE FILM DUE TO A ROTATING OR DITHERING LASER BEAM .....</b>	<b>35</b>
<b>A.</b>	<b>THE CRANK-NICOLSON METHOD .....</b>	<b>35</b>
<b>B.</b>	<b>THE FAST FOURIER TRANSFORM (FFT) METHOD .....</b>	<b>46</b>
<b>C.</b>	<b>COMSOL.....</b>	<b>48</b>
<b>D.</b>	<b>COMPARISON ON A MODEL PROBLEM .....</b>	<b>49</b>
<b>E.</b>	<b>REAL PROBLEM SIMULATION (STEEL AISI 4340) .....</b>	<b>51</b>
<b>V.</b>	<b>NUMERICAL SOLUTION FOR A TRANSIENT, THREE-DIMENSIONAL TEMPERATURE DISTRIBUTION IN A FINITE SOLID DUE TO A ROTATING OR DITHERING LASER BEAM.....</b>	<b>55</b>
<b>VI.</b>	<b>CONCLUSIONS AND FUTURE WORK.....</b>	<b>63</b>
	<b>APPENDIX A. CRANK-NICOLSON CODE FOR 1-D SIMULATION .....</b>	<b>65</b>
	<b>APPENDIX B. FFT CODE FOR 1-D SIMULATION .....</b>	<b>67</b>
	<b>APPENDIX C. COMSOL CODE FOR 1-D SIMULATION.....</b>	<b>69</b>
	<b>APPENDIX D. CRANK-NICOLSON CODE FOR 2-D SIMULATION .....</b>	<b>75</b>
	<b>APPENDIX E. FFT CODE FOR 2-D SIMULATION .....</b>	<b>79</b>
	<b>APPENDIX F. COMSOL CODE FOR 2-D SIMULATION .....</b>	<b>81</b>
	<b>APPENDIX G. COMSOL CODE FOR 3-D SIMULATION .....</b>	<b>89</b>
	<b>LIST OF REFERENCES.....</b>	<b>99</b>
	<b>INITIAL DISTRIBUTION LIST .....</b>	<b>101</b>

THIS PAGE INTENTIONALLY LEFT BLANK

## LIST OF FIGURES

Figure 1.	Different transversal modes in a laser spot [From 1].....	4
Figure 2.	Dithering laser beam on a 1-D rod.....	5
Figure 3.	Effective radius of laser beam, $d$ as a multiple of standard deviation $\delta$ .....	11
Figure 4.	Domain expansion: cosine expansion to FFT .....	13
Figure 5.	Implicit Crank-Nicolson scheme [From 7]. .....	17
Figure 6.	Discretization of the 1-D rod .....	18
Figure 7.	1-D Crank-Nicolson method.....	24
Figure 8.	1-D FFT method .....	28
Figure 9.	1-D COMSOL.....	29
Figure 10.	Relative error plot in 1-D Crank-Nicolson and COMSOL.....	30
Figure 11.	Temperature rise on the material of steel AISI 4340 using the 1-D Crank-Nicolson method .....	32
Figure 12.	Temperature rise on the material of steel AISI 4340 using 1-D COMSOL ....	33
Figure 13.	1-D maximum temperature rise of steel AISI 4340 versus the frequency of the dithering laser beam.....	33
Figure 14.	1-D temperature change with time at a fixed point $x=0.75$ of steel AISI 4340 with dithering laser beam period=0.1s.....	34
Figure 15.	The maximum temperature rise of steel AISI 4340 versus frequency with two different heat source $I_0$ .....	34
Figure 16.	2-D schematic of the laser beam and the finite work piece .....	35
Figure 17.	Laser beam creates heat source as a Gaussian distribution [After 2]. .....	36
Figure 18.	Discretization of a 2-D rectangular region.....	36
Figure 19.	Putting the temperature of each point in a vector form $\vec{u}^k$ after discretization .....	38
Figure 20.	Snapshots of the temperature rise on a film of a model problem induced by a rotating Gaussian beam using the 2-D Crank-Nicolson method.....	46
Figure 21.	Snapshots of the temperature rise on a film of a model problem induced by a rotating Gaussian beam using 2-D FFT method in 3-D view .....	48
Figure 22.	Model problem comparison: (a) COMSOL (b) MATLAB FFT 2 method ....	49
Figure 23.	Efficiency plot: Crank-Nicolson 2-D method versus FFT 2-D method .....	50
Figure 24.	Relative error plot in 2-D FFT method and COMSOL .....	50
Figure 25.	Snapshots of the temperature rise on a Steel AISI 4340 film induced by a rotating Gaussian beam using COMOSL with period=1s .....	52
Figure 26.	2-D temperature change with time at a fixed point of Steel AISI 4340 with rotating beam period=0.1s .....	53
Figure 27.	2-D maximum temperature rise of Steel AISI 4340 versus the frequency of the rotating laser beam.....	53
Figure 28.	Snapshots of the temperature rise on a Steel AISI 4340 solid induced by a rotating Gaussian beam using COMSOL with period=1s .....	56
Figure 29.	3-D temperature change as a function of time at a fixed point of steel AISI 4340 with rotating laser beam period=1s.....	57

Figure 30.	Maximum temperature as a function of time with rotating laser beam period =1s .....	57
Figure 31.	3-D temperature rise at different layers from $z=0$ to $z=1$ .....	58
Figure 32.	Maximum temperature at different depth (a) $z=1$ top surface (b) $z=0.99$ (c) $z=0.95$ (d) $z=0.90$ .....	58
Figure 33.	Snapshots of the temperature rise on a Steel AISI 4340 solid induced by a rotating Gaussian beam using COMSOL with period=0.1s .....	59
Figure 34.	3-D temperature change as a function of time at a fixed point of Steel AISI 4340 with rotating laser beam period=0.1 .....	60
Figure 35.	Maximum temperature as a function of time with rotating laser beam period =0.1s .....	60
Figure 36.	3-D maximum temperature rise of Steel AISI 4340 versus the frequency of the rotating laser beam .....	61

## LIST OF TABLES

Table 1.	1-D rod MATLAB input parameters for dithering laser beam .....	23
Table 2.	Thermal property of Steel AISI 4340 .....	31
Table 3.	1-D dithering laser input on Steel AISI 4340 .....	31
Table 4.	2-D film MATLAB input parameters for rotating laser beam.....	45
Table 5.	2-D rotating laser input on Steel AISI 4340. ....	51
Table 6.	3-D rotating laser input on Steel AISI 4340 .....	55

THIS PAGE INTENTIONALLY LEFT BLANK

## **ACKNOWLEDGMENTS**

First and foremost, I would like to acknowledge my wife and son, Alice and Edison, for their endless family support. I could not finish this work without them.

The success of this thesis and my Master's Degree completion at the Naval Postgraduate School are due to the generosity and tremendous attitude of the Department of Physics and the Department of Applied Mathematics.

Particularly, I owe a great deal of gratitude to my thesis advisor, Dr. Hong Zhou, whose expertise in numerical analysis and mathematical modeling has enabled me to accomplish this work. I would also thank my thesis co-advisor, Dr. Karunasiri, as well as Dr. John Dunec and Dr. Jan Wang from COMSOL Inc., for their constant willingness to enlighten me on COMSOL.

Finally, I would like to dedicate this thesis to my father, who passed away on Oct. 13, 2010, two months before I finished this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK



# I. INTRODUCTION

Laser-induced heating has been widely studied [1, 2, 3, 4]. Most of these studies are restricted to a moving/scanning laser beam. Recently, the effect of a rotating/dithering laser beam on the temperature rise of a semi-infinite domain has been studied [5]. However, in many realistic problems the geometry of an object is finite. So, in this thesis, we extend the previous studies to a finite domain and pinpoint the effect of a rotating/dithering laser beam on the temperature rise of a finite object.

The laser beam can be considered as a moving disc heat source with a Gaussian distribution of heat intensity. The analytical solutions can be used to determine the temperature rise distribution in and around the laser beam heat source on the work surface as well as with respect to depth at all domains close to the heat source. The analytical and numerical solutions from mathematical approaches provide a better appreciation of the physical relationships between the relevant laser parameters and the thermal properties of the work piece.

The main purpose of this thesis is to investigate the relationship between the rotating/dithering laser beam speed and the maximum temperature rise induced by the laser beam on a finite solid. In Chapters II and III, we present the analytical and numerical methods for obtaining a transient temperature distribution with various methods, which include the eigenfunction expansion method, the Crank-Nicolson method and the Fast Fourier Transform method (FFT). In Chapters III and IV, numerical solutions for one-dimensional (1-D) and two-dimensional (2-D) nonhomogeneous heat equations from MATLAB and COMSOL are given. We compare different methods and investigate the relative errors of numerical solutions obtained in MATLAB and COMSOL. In Chapter V, we have employed COMSOL to solve the three-dimensional (3-D) problem.

The objective of this thesis is to provide insights on developing objects of adequate rotating speed against direct energy weapons (DEWs). The simulation results, especially the quantitative relationship between maximum temperature rise and

rotating/dithering speed, can be used as a blueprint to adjust the speed of rotation of the target against DEWs to prevent the temperature rise from reaching the melting point of the target.

## II. ANALYTICAL SOLUTION FOR A TRANSIENT TEMPERATURE DISTRIBUTION DUE TO A ROTATING OR DITHERING LASER BEAM IN A FINITE SOLID: EIGEN-FUNCTION EXPANSION METHOD

Consider a laser beam source used to heat a 1-D finite rod. A 1-D heat equation can be written as [1]

$$\frac{\partial u}{\partial t} = \alpha_T \frac{\partial^2 u}{\partial x^2} + \frac{\alpha_T}{K_T} q(x, t) \quad 0 < x < L_x \quad (\text{Eq. 2.1})$$

where  $u(x, t)$  is the temperature rise with respect to the ambient temperature,  $q(x, t)$  is the heat source created by laser beam,  $\alpha_T$  is the thermal diffusivity with units  $\left[ \frac{m^2}{s} \right]$  and  $K_T$  is the thermal conductivity with units  $\left[ \frac{W}{m \cdot k} \right]$ .

Two assumptions are made in this analytical approach. First, heat losses by radiation are negligible as compared to the intensity of the incident laser beam. Second, thermal properties, such as thermal conductivity  $K_T$  and thermal diffusivity  $\alpha_T$ , are considered constant and evaluated at an average temperature. The second assumption makes this heat equation a linear problem.

For this 1-D rod, we impose insulating boundary conditions (BCs)

$\frac{\partial u}{\partial x}(0, t) = \frac{\partial u}{\partial x}(L_x, t) = 0$ , which assumes that no energy escapes into the air at the air/material interface. This is a good approximation for most materials under consideration, as heat flow by conduction through the material far exceeds the heat loss by radiation or convection at the air/material interface.

The initial condition (IC) is  $u(x, 0) = 0$ , which reflects the fact that there is no temperature rise with respect to the ambient temperature before the laser hits the rod.

The heat source  $q(x,t)$  is considered as a laser beam of temporal continuous wave (CW) and spatially modeled by a Gaussian distribution, which corresponds to the theoretical  $TEM_{00}$  mode of the laser. The term  $TEM$  comes from the acronym “transverse electromagnetic mode.” The subscript of TEM specifies the number of nodes generated by a slight misalignment of the mirrors located in the laser cavity [1, 6]. Figure 1 shows some transverse modes and the simplest modes,  $TEM_{00}$  is used in this thesis; the heat source generated by the laser beam is a Gaussian distribution.

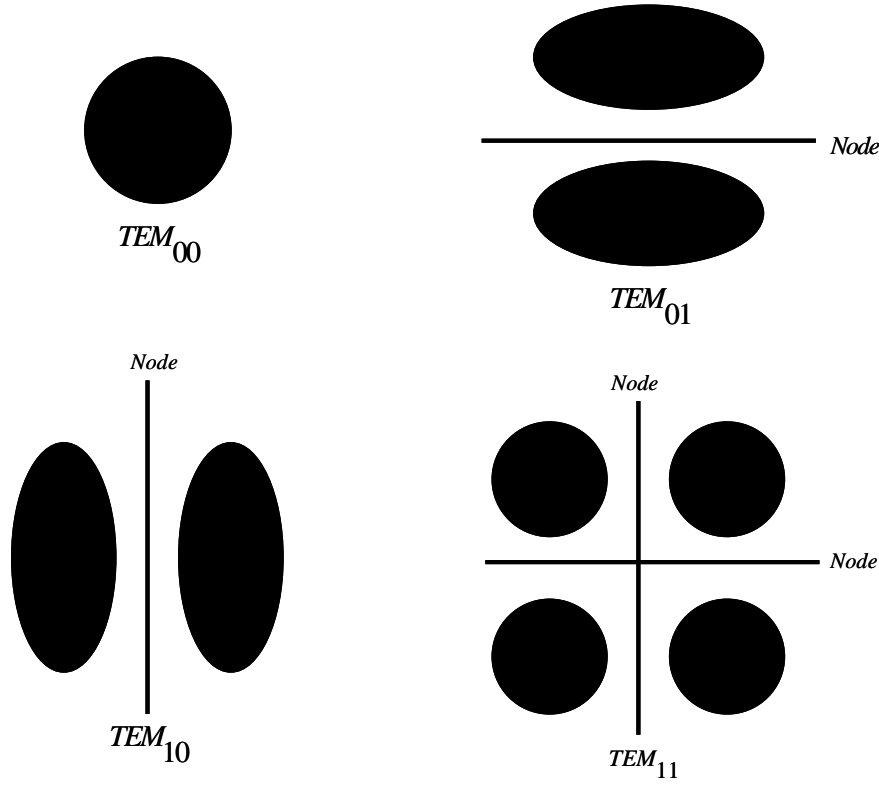


Figure 1. Different transversal modes in a laser spot [From 1]

For a dithering laser beam in  $TEM_{00}$  mode, the heat source  $q(x,t)$  can be written as

$$q(x,t) = I_0 e^{-\frac{k}{r_0^2}(x-x_c(t))^2}, \quad x_c(t) = x_0 + a \sin \frac{2\pi t}{T}, \quad x_0 = \frac{L_x}{2}$$

Here,  $x_c(t)$  is the position of the dithering Gaussian beam and  $x_0$  is the initial position of the laser beam. In the above formula, the center point of the rod is picked.  $r_0$  is the effective radius of the laser beam.  $I_0$  is the intensity of the laser beam at the center of the heat spot after it is absorbed by the material.  $k$  is a constant used for the Gaussian model [2]. Figure 2 depicts a dithering laser beam shining on a 1-D rod.

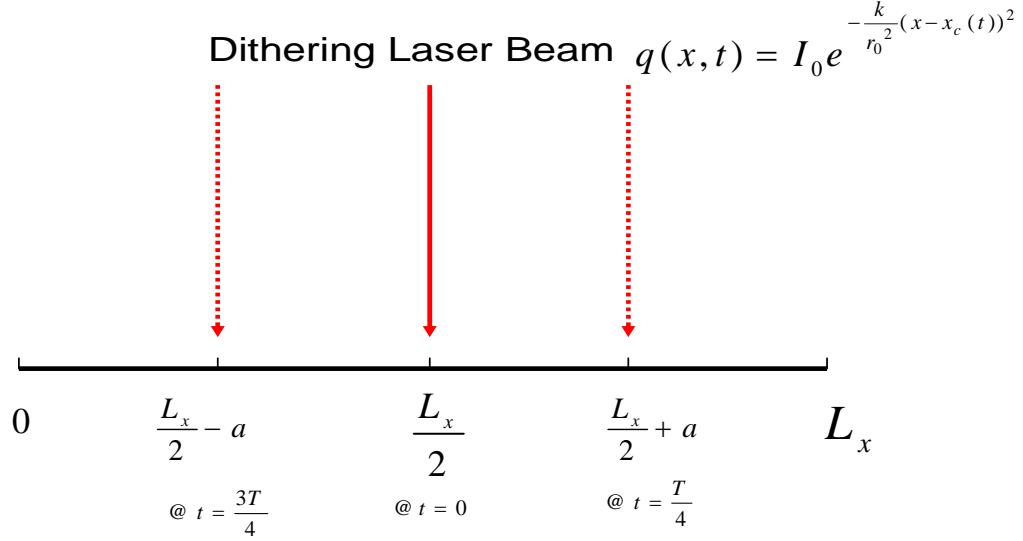


Figure 2. Dithering laser beam on a 1-D rod

We will apply the eigenfunction expansion method to solve this 1-D heat equation. Generally speaking, the method of eigenfunction expansions consists of building up the solution of a boundary value problem as a sum of eigenfunctions of a Helmholtz problem. To begin with, consider a simple 1-D mathematical model in the Cartesian coordinate for solving the linear and homogeneous heat equation with initial conditions (ICs) and boundary Conditions (BCs) described above where no heat source  $q(x, t)$  is involved, the 1-D heat equation becomes:

$$\frac{\partial u}{\partial t} = \alpha_T \nabla^2 u.$$

To make the derivation process simpler, set  $\alpha_T \equiv 1$ . The method of separation of variables (SOV) is applied in solving this homogeneous heat equation [1]. We will move on to a nonhomogeneous solution with heat source  $q(x,t)$  by applying the method of eigenfunction expansion.

It is intuitional that the temperature  $u$  is a function of position and time, therefore  $u = u(x,t)$ . In the method of SOV, we attempt to determine solutions in the product form,

$$u(x,t) = X(x)T(t) \quad (\text{Eq. 2.2})$$

where  $X(x)$  is a function of space  $x$  alone and  $T(t)$  is a function of time  $t$  alone. This SOV reduces a partial differential equation (PDE) to several ordinary differential equations (ODEs).

For a heat equation without heat source:

$$\frac{\partial u(x,t)}{\partial t} = \nabla^2 u(x,t) \quad (\text{Eq. 2.3})$$

We substitute the assumed product form of Eq. 2.2 into this heat equation and get

$$X(x)T'(t) = X''(x)T(t)$$

We can actually “separate variables” by dividing both sides of this equation by  $X(x)T(t)$  to obtain

$$\frac{T'}{T} = \frac{X''}{X} \quad (\text{Eq. 2.4})$$

Now the variables have been “separated” in the sense that the left-hand side of the equation is only a function of time  $t$  and the right-hand side function is a function of space  $x$ . Since the variables  $t$  and  $x$  are independent of each other, the only way to get equality is to have the functions on both sides of (Eq. 2.4) constant and equal. Thus,

$$\frac{T'}{T} = \frac{X''}{X} = -\lambda$$

where  $\lambda$  is an arbitrary constant called the separation constant. Later we will explain why this separation constant is a negative value with  $\lambda > 0$ . Since we treat the laser beam as a heat source, and there is no heat source involved yet in solving the homogeneous equation, it is intuitive that the temperature decreases as a function of time, so

$$\frac{T'(t)}{T(t)} = -\lambda < 0 \quad (\text{Eq. 2.5})$$

(Eq. 2.5) is a first-order linear homogeneous differential equation with constant coefficient. This ODE can be solved directly by seeking exponential solutions,  $T = e^{rt}$ . By substitution, the characteristic polynomial is  $r = -\lambda$ . Therefore, the general solution of (Eq. 2.5) is

$$T(t) = ce^{-\lambda t} \quad (\text{Eq. 2.6})$$

Where  $c$  is an arbitrary constant. The time-dependent solution is a simple exponential. Recall that  $\lambda$  is the separation constant, which for the moment is arbitrary; however, eventually we will find out that only certain values of  $\lambda$  are allowable. If  $\lambda > 0$ , the solution exponentially decays as  $t$  increases because of the negative sign; if  $\lambda < 0$ , the solution exponentially increases; if  $\lambda = 0$ , the solution remains constant in time. Therefore,  $\lambda > 0$  gives a reasonable solution since temperature decreases as time goes by.

Our next move is to consider the right-hand side of (Eq. 2.4):

$$\frac{X''}{X} = -\lambda \quad (\lambda > 0) \quad (\text{Eq. 2.7})$$

This is a second-order, constant coefficient homogeneous ODE, substituting  $X(x) = e^{rx}$  into (Eq. 2.7) yields the characteristic equation  $r^2 + \lambda = 0$ .

Since  $\lambda > 0$ , exponential solutions have imaginary exponents,  $X(x) = e^{\pm i\sqrt{\lambda}x}$ . In this case, the solution oscillates. Recalling Euler's formula,  $e^{i\theta} = \cos(\theta) + i\sin(\theta)$ , so the choices  $\cos(\sqrt{\lambda}x)$  and  $\sin(\sqrt{\lambda}x)$  are made to get the desired real number solutions. The general solution for (Eq. 2.7) is:

$$X(x) = A\cos(\sqrt{\lambda}x) + B\sin(\sqrt{\lambda}x) \quad (\text{Eq. 2.8})$$

This is an arbitrary linear combination of two independent solutions where A and B are just arbitrary constants.

Both ends of the 1-D rod are insulated; the BCs on  $x$  direction states:

$$\left. \frac{\partial u}{\partial X} \right|_{x=0} = 0 \quad \text{and} \quad \left. \frac{\partial u}{\partial X} \right|_{x=L_x} = 0$$

This is equivalent to

$$X'(0) = 0 \quad \text{and} \quad X'(L_x) = 0 \quad (\text{BCs})$$

The derivative of (Eq. 2.8) is

$$X'(x) = -A\sqrt{\lambda}\sin(\sqrt{\lambda}x) + B\sqrt{\lambda}\cos(\sqrt{\lambda}x) \quad (\text{Eq. 2.9})$$

We can plug the first BC  $X'(0) = 0$  into (Eq. 2.9) to obtain the solution  $X'(0) = 0$  which implies that  $B=0$ . So  $X(x)$  is a function of cosine only since the sine term vanishes. Another BC implies that  $A\sin(\sqrt{\lambda}L_x) = 0$ . To avoid the trivial solution  $X(0) = 0$ , we take  $A=1$ , which forces  $\sin(\sqrt{\lambda}L_x) = 0$ . Since the sin function vanishes at the integer multiples of  $\pi$ , we conclude that

$$\sqrt{\lambda} = \sqrt{\lambda_n} = \frac{n\pi}{L_x}, \quad n = 0, \pm 1, \pm 2, \dots$$

and so



$$X = X_n(x) = \cos\left(\frac{n\pi x}{L_x}\right), \quad n = 0, 1, 2, \dots$$

Note that for negative values of  $n$ , we obtain the same solution; hence, solutions corresponding to negative  $n$ 's may be discarded without loss. These  $X_n(x)$  are from the eigenfunction of the 1-D heat equation. We now go back to (Eq. 2.6) and substitute

$$\lambda = \left(\frac{n\pi}{L_x}\right)^2 \text{ to get}$$

$$T_n(t) = c_n e^{-\left(\frac{n\pi}{L_x}\right)^2 t}$$

Now we return to (Eq. 2.1). We solve this problem using the method of eigenfunction expansion, and hence start by assuming that  $u$  has an expansion in terms of the eigenfunctions  $X_n(x)$ . Thus,

$$u(x, t) = \underbrace{\sum_n u_n(t) \cos\left(\frac{n\pi x}{L_x}\right)}_{\text{eigenfunction expansion}} \quad (\text{Eq. 2.10})$$

where the coefficients  $u_n(t)$  are functions of  $t$ . We now expand  $q$  as

$$q(x, t) = \sum_n q_n(t) X_n(x) \quad (\text{Eq. 2.11})$$

After substituting (Eq. 2.10) and (Eq. 2.11) into the 1-D inhomogeneous heat equation (Eq. 2.1), we obtain

$$\sum_n u_n'(t) X_n(x) = \alpha_T \sum_n u_n(t) \left[ -\left(\frac{n\pi}{L_x}\right)^2 \right] X_n(x) + \frac{\alpha_T}{K_T} \sum_n q_n(t) X_n(x).$$

This yields the differential equation of the coefficients  $u_n(t)$ ,

$$u_n'(t) = \alpha_T \left[ -\left(\frac{n\pi}{L_x}\right)^2 \right] u_n(t) + \frac{\alpha_T}{K_T} q_n(t) \quad (\text{Eq. 2.12})$$

The initial condition for this equation is  $u_n(t=0)=0$ .

Now the problem is to find  $q_n(t)$ . We can use the property of orthogonality of eigenfunctions. By multiplying eigenfunction  $X_n(x)$  which is  $\cos(\frac{n\pi x}{L_x})$  and then integrating on both sides of Eq. 2.11, we get

$$\begin{aligned}\int_0^{L_x} q(x,t) X_n(x) dx &= \int_0^{L_x} \sum_n q_n(t) X_n(x) \cdot X_n(x) dx = q_n(t) \int_0^{L_x} X_n^2(x) dx \\ &= q_n(t) \int_0^{L_x} \frac{1 + \cos(\frac{2n\pi x}{L_x})}{2} dx = q_n(t) \frac{L_x}{2}\end{aligned}$$

$$\text{Therefore, } q_n(t) = \frac{2}{L_x} \int_0^{L_x} q(x,t) X_n(x) dx = \frac{2}{L_x} \int_0^{L_x} I_0 e^{-\frac{k}{r_0^2}(x-x_c(t))^2} \cos(\frac{n\pi x}{L_x}) dx$$

To obtain  $q_n(t)$  is very expansive because the  $x-x_c(t)$  term makes the integration time-dependent. Here we use change of variables to make this calculation easier,

$$\begin{aligned}\tilde{x} &= x - x_c(t) \\ d\tilde{x} &= dx\end{aligned}$$

thus,

$$q_n(t) = \frac{2}{L_x} \int_{-x_c(t)}^{L_x-x_c(t)} I_0 e^{-\frac{k}{r_0^2}\tilde{x}^2} \cos(\frac{n\pi}{L_x}(\tilde{x} + x_c(t))) d\tilde{x}$$

We introduce a constant  $d$ , which is the effective radius of the laser beam, and

$d$  is about 3 to 5 times of the standard deviation  $\delta$  of  $I_0 e^{-\frac{k}{r_0^2}\tilde{x}^2}$ . Instead of taking integral from  $\tilde{x} = -x_c(t)$  to  $L_x - x_c(t)$ , using the assumption above to change the integral from  $\tilde{x} = -d$  to  $\tilde{x} = d$ , since the laser beam is a very concentrated Gaussian distribution, so taking integral with respect to a large range is unnecessary; we only need to integrate within the effective laser range which is from  $-d$  to  $d$ . Figure 3 shows the effective radius of the laser beam.

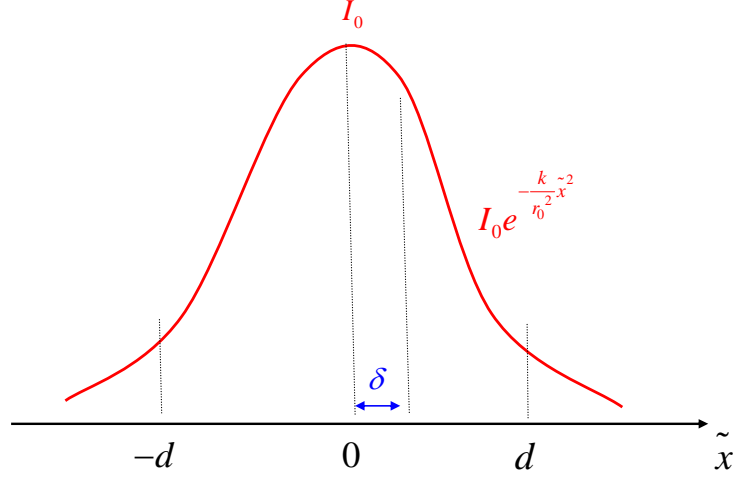


Figure 3. Effective radius of laser beam,  $d$  as a multiple of standard deviation  $\delta$

We also recall that the trigonometry formula of cosine in sum and difference formula,

$$\cos(u \pm v) = \cos(u) \cos(v) \mp \sin(u) \sin(v)$$

then  $q_n(t)$  becomes

$$\begin{aligned} q_n(t) = & \frac{2}{L_x} \left( \int_{-d}^d I_0 e^{-\frac{k}{r_0^2} \tilde{x}^2} \cos\left(\frac{n\pi \tilde{x}}{L_x}\right) d\tilde{x} \right) \cos\left(\frac{n\pi x_c(t)}{L_x}\right) \\ & - \frac{2}{L_x} \left( \int_{-d}^d I_0 e^{-\frac{k}{r_0^2} \tilde{x}^2} \sin\left(\frac{n\pi \tilde{x}}{L_x}\right) d\tilde{x} \right) \sin\left(\frac{n\pi x_c(t)}{L_x}\right) \end{aligned}$$

The integral in the second term is zero since the integrand function is an odd function and the integration interval is  $[-d, d]$ . Thus,  $q_n(t)$  simplifies to

$$q_n(t) = \frac{2}{L_x} \left( \int_{-d}^d I_0 e^{-\frac{k}{r_0^2} \tilde{x}^2} \cos\left(\frac{n\pi \tilde{x}}{L_x}\right) d\tilde{x} \right) \cos\left(\frac{n\pi x_c(t)}{L_x}\right)$$

Once  $q_n(t)$  is found, we have to compute  $u_n(t)$  using the method of integrating factor.

Recall the method of integrating factor in solving first-order ODE:

$$y'(t) + a \cdot y(t) = r(t)$$

Here  $a$  is a constant and  $r(t)$  is a function of time. The first step of integrating factor is to multiply  $e^{at}$  on both sides of this equation:

$$e^{at} \cdot [y'(t) + a \cdot y(t)] = e^{at} \cdot r(t)$$

This equation is equivalent to

$$\frac{d}{dt} [e^{at} \cdot y(t)] = e^{at} \cdot r(t)$$

We can integrate on both sides of the equation and get

$$e^{at} \cdot y(t) = y(0) + \int_{t=0}^t e^{as} \cdot r(s) ds$$

where  $s$  is a dummy variable. We multiply  $e^{-at}$  to obtain

$$y(t) = y(0) \cdot e^{-at} + e^{-at} \cdot \int_{t=0}^t e^{at} \cdot r(t) dt$$

Now, we can compute  $u_n(t)$  of Eq. 2.12 use the integrating factor method to get

$$\begin{aligned} u_n'(t) &= \alpha_T \left[ - \left( \frac{n\pi}{L_x} \right)^2 \right] u_n(t) + \frac{\alpha_T}{K_T} q_n(t) \\ \Rightarrow \frac{d}{dt} \left[ e^{\alpha_T \left( \frac{n\pi}{L_x} \right)^2 t} u_n(t) \right] &= e^{\alpha_T \left( \frac{n\pi}{L_x} \right)^2 t} \frac{\alpha_T}{K_T} q_n(t) \\ \Rightarrow e^{\alpha_T \left( \frac{n\pi}{L_x} \right)^2 t} u_n(t) &= \int_0^t e^{\alpha_T \left( \frac{n\pi}{L_x} \right)^2 s} \frac{\alpha_T}{K_T} q_n(s) ds \\ \Rightarrow u_n(t) &= e^{-\alpha_T \left( \frac{n\pi}{L_x} \right)^2 t} \cdot \left[ \int_0^t e^{\alpha_T \left( \frac{n\pi}{L_x} \right)^2 s} \frac{\alpha_T}{K_T} q_n(s) ds \right] \end{aligned}$$

where the initial condition  $u_n(0)$  has been applied.

$$\text{Once } u_n(t) \text{ is calculated, we can find } u(x,t) = \sum_n u_n(t) X_n(x) = \sum_n u_n(t) \cos\left(\frac{n\pi x}{L_x}\right)$$

After we conduct several experiments in MATLAB, we find that the direct summation is very slow; the computational cost of this eigenfunction expansion method is extremely high in a 3-D nonhomogeneous heat equation problem. An alternative way to compute the series is to take advantage of the Fast Fourier Transform (FFT). In order to do so, we first change the cosine form to a combination of exponential forms.

$$\cos\left(\frac{n\pi x}{L_x}\right) = \frac{e^{i\frac{n\pi x}{L_x}} + e^{-i\frac{n\pi x}{L_x}}}{2}$$

Then we even extend  $u(x,t)$  from the domain  $(0, L_x)$  to  $(-L_x, 0)$  and make it periodic with period  $2L_x$ . Figure 4 shows how to expand the domain in FFT method.

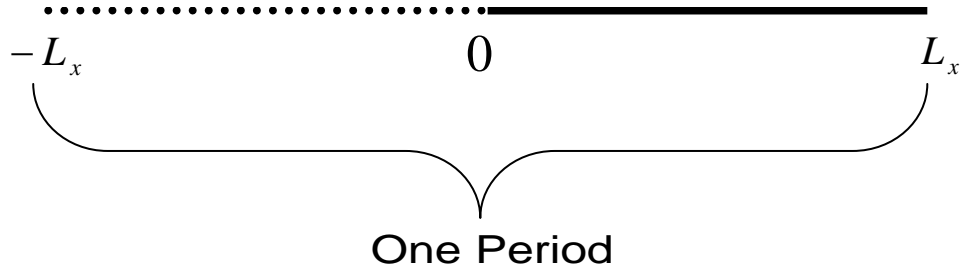


Figure 4. Domain expansion: cosine expansion to FFT

So, we rewrite  $u$  as

$$u(x,t) = \sum_{n=0}^{\infty} u_n(t) \left( \frac{e^{i \frac{n\pi x}{L_x}} + e^{-i \frac{n\pi x}{L_x}}}{2} \right)$$

$$= \sum_{n=-\infty}^{\infty} c_n(t) \cdot e^{i \frac{n\pi x}{L_x}} \quad \text{where} \quad c_n(t) = \begin{cases} \frac{u_n(t)}{2}, & n \geq 0 \\ \frac{u_{-n}(t)}{2}, & n < 0 \end{cases}$$

and then we apply FFT. We will provide more details on how to compute the solution with FFT in Chapter III.

### III. NUMERICAL SOLUTION FOR A TRANSIENT, ONE-DIMENSIONAL TEMPERATURE DISTRIBUTION IN A FINITE ROD DUE TO A DITHERING LASER BEAM

In this section, we review some numerical methods for solving the 1-D nonhomogeneous heat equation.

#### A. THE CRANK-NICOLSON METHOD

The Crank-Nicolson method is an implicit, unconditionally stable numerical method used to solve the heat equation. It is second-order accurate in time and second-order accurate in space. In 1947, Crank and Nicolson suggested an alternative way to utilize centered differences scheme [7]. To illustrate the Crank-Nicolson scheme, we start with centered difference and return to the Taylor series expansion for  $f(x_0 + \Delta x)$  and  $f(x_0 - \Delta x)$ .

$$f(x + \Delta x) = f(x) + \frac{\Delta x}{1!} f'(x) + \frac{(\Delta x)^2}{2!} f''(x) + \frac{(\Delta x)^3}{3!} f^{(3)}(x) + \dots \quad (\text{Eq. 3.1})$$

$$f(x - \Delta x) = f(x) - \frac{\Delta x}{1!} f'(x) + \frac{(\Delta x)^2}{2!} f''(x) - \frac{(\Delta x)^3}{3!} f^{(3)}(x) + \dots \quad (\text{Eq. 3.2})$$

Subtracting (Eq. 3.1) from (Eq. 3.2) yields

$$f(x + \Delta x) - f(x - \Delta x) = 2\Delta x f'(x) + \frac{2}{3!} (\Delta x)^3 f^{(3)}(x) + \dots$$

We can get  $f'(x)$  by reorganizing this equation

$$f'(x) = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x} - \underbrace{\frac{1}{6} (\Delta x)^2 f^{(3)}(x) + \dots}_{=O(\Delta x)^2}$$

This leads to the first-order centered difference approximation

$$f'(x) \approx \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x}$$

This approximation is with truncation error  $O(\Delta x)^2$ , an order of  $(\Delta x)^2$  and the approximation is accurate and consistent; the truncation error vanishes as  $\Delta x \rightarrow 0$ .

By adding Eq. 3.1 and Eq. 3.2, we have

$$f(x + \Delta x) + f(x - \Delta x) = 2f(x) + (\Delta x)^2 f''(x) + \frac{2}{4!} (\Delta x)^4 f^{(4)}(x) + \dots$$

Again, we can get the second-order derivative  $f''(x)$  by reorganizing this equation

$$f''(x) = \frac{f(x + \Delta x) - 2f(x) + f(x - \Delta x)}{(\Delta x)^2} - \underbrace{\frac{(\Delta x)^2}{12} f^{(4)}(x) + \dots}_{O(\Delta x)^2}$$

and this leads to the second-order centered difference approximation

$$f''(x) \approx \frac{f(x + \Delta x) - 2f(x) + f(x - \Delta x)}{(\Delta x)^2}$$

After we understand the second-order difference approximation, we can move on to the Crank-Nicolson scheme. The forward difference in time of temperature  $T = T(x, t)$  can be written as:

$$\frac{\partial T}{\partial t} \approx \frac{T(t + \Delta t) - T(t)}{\Delta t}$$

The Crank-Nicolson method may be interpreted as the centered difference around  $t + \frac{\Delta t}{2}$ . The error in approximating  $\frac{\partial T}{\partial t}(t + \frac{\Delta t}{2})$  is  $O(\Delta t)^2$  [7]. Thus, we discretize the second derivative at  $t + \frac{\Delta t}{2}$  with a centered difference scheme. Since this involves functions evaluated at this in-between time, we take the average at  $t$  and  $t + \Delta t$ . This yields the Crank-Nicolson scheme for 1-D homogeneous heat equation  $\frac{\partial T}{\partial t} = \alpha_T \frac{\partial^2 T}{\partial x^2}$ ,

$$\frac{T_i^{k+1} - T_i^k}{\Delta t} = \alpha_T \left[ \frac{T_{i+1}^k - 2T_i^k + T_{i-1}^k}{\Delta x^2} + \frac{T_{i+1}^{k+1} - 2T_i^{k+1} + T_{i-1}^{k+1}}{\Delta x^2} \right] \quad (\text{Eq. 3.3})$$



We introduce the shorthand notation  $T(x_i, t_k) = T_i^k$ , which is a matrix notation used to discretize  $x$  and  $t$ , because the Crank-Nicolson scheme involves six stencil points rather than a simpler stable method, and three of which are at the advanced time, as shown in Figure 5. We cannot directly march forward in time. Instead, we are left with a traditional system of equations of the form  $A\vec{x} = \vec{b}$ , where  $\vec{x}$  is the solution at the  $(k+1)st$  time level and  $\vec{b}$  depends on the solution at the  $k-th$  time level. Of course, it is not just any  $A$ , the matrix we have has a very special structure which is to be shown later.

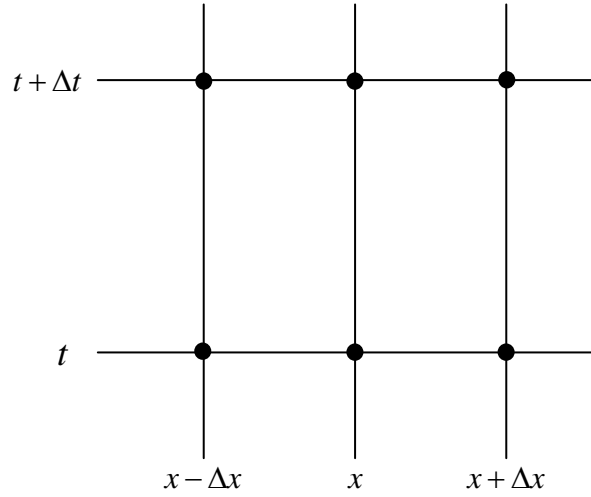


Figure 5. Implicit Crank-Nicolson scheme [From 7].

Now consider a 1-D nonhomogeneous heat equation with a heat source  $f(x, t)$ :

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} + f(x, t) \quad , \quad 0 \leq x \leq 1$$

The BCs are insulated at both ends of the rod:

$$\left. \frac{\partial T}{\partial x} \right|_{x=0} = \left. \frac{\partial T}{\partial x} \right|_{x=1} = 0$$

The IC assumes that there is no temperature rise with respect to the ambient temperature:

$$T(x, 0) = 0$$

In this thesis, we study the case where the heat source is a dithering 1-D Gaussian beam:

$$f(x, t) = I_0 e^{-\frac{(x-x_c(t))^2}{r_0^2}}, \quad x_c(t) = x_0 + a \sin \frac{2\pi t}{T}, \quad x_0 = \frac{L_x}{2}$$

where  $I_0$  is the intensity of the laser beam. As before, we discretize this 1-D rod into  $n$  parts, as depicted in Figure 6:

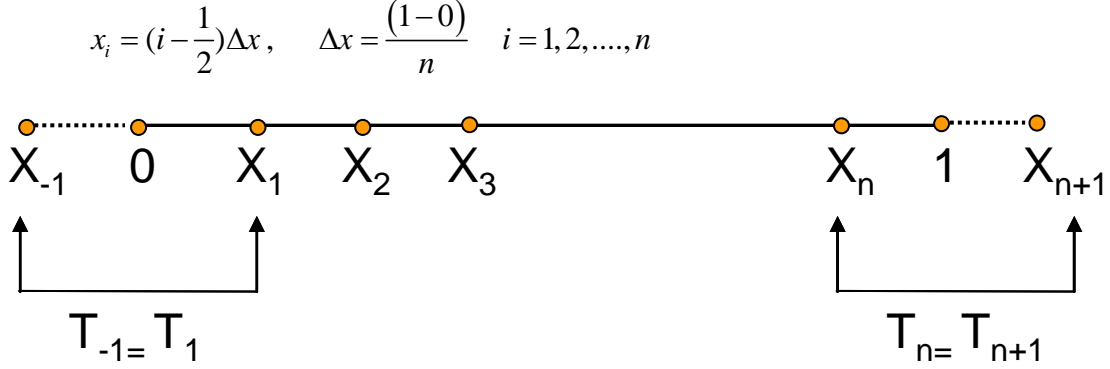


Figure 6. Discretization of the 1-D rod

Here in the Crank-Nicolson scheme and later in the matrix notation, the BCs are extended into ghost grid points  $x_{-1}$  and  $x_{n+1}$ . However, in this finite difference method, the BCs link temperature at  $x_{-1}$  to the temperature at  $x_1$  and the temperature at  $x_{n+1}$  to the temperature at  $x_n$ . More specifically, the BCs are

$$\left. \frac{\partial T}{\partial x} \right|_{x=0} \approx \frac{T(x_1, t_k) - T(x_{-1}, t_k)}{\Delta x} = 0 \Rightarrow T_1^k = T_{-1}^k$$

$$\left. \frac{\partial T}{\partial x} \right|_{x=1} \approx \frac{T(x_{n+1}, t_k) - T(x_n, t_k)}{\Delta x} = 0 \Rightarrow T_{n+1}^k = T_n^k \text{ for all } k \geq 1$$

Applying the Crank-Nicolson scheme to the heat equation with a heat source  $f(x, t)$ , we obtain

$$\frac{T_i^{k+1} - T_i^k}{\Delta t} = \frac{\alpha}{2} \left[ \frac{T_{i+1}^k - 2T_i^k + T_{i-1}^k}{\Delta x^2} + \frac{T_{i+1}^{k+1} - 2T_i^{k+1} + T_{i-1}^{k+1}}{\Delta x^2} \right] + \frac{1}{2} [f(x_i, t_k) + f(x_i, t_{k+1})]$$

$$i = 1, 2, \dots, n-1, n \quad k = 0, 1, 2, \dots \quad t_k = k\Delta t$$

Here,  $\{T_i^k\}$  are known,  $\{T_i^{k+1}\}$  are to be found. We start with a simple case  $i=1$  and rearrange this equation:

$$T_1^{k+1} = T_1^k + \frac{\alpha\Delta t}{2} \left[ \frac{T_2^k - 2T_1^k + \mathbf{T}_{-1}^k}{\Delta x^2} + \frac{T_2^{k+1} - 2T_1^{k+1} + \mathbf{T}_{-1}^{k+1}}{\Delta x^2} \right] + \frac{\Delta t}{2} [f(x_1, t_k) + f(x_1, t_{k+1})]$$

where  $\mathbf{T}_{-1}^k = T_1^k$  and  $\mathbf{T}_{-1}^{k+1} = T_1^{k+1}$  because of the BCs. After changing some orders, we put  $T_i^{k+1}$  on the left-hand side of the equation and  $T_i^k$  on the right-hand side; we get

$$T_1^{k+1} - \frac{\alpha\Delta t}{2\Delta x^2} T_2^{k+1} + \frac{\alpha\Delta t}{2\Delta x^2} T_1^{k+1} = T_1^k + \frac{\alpha\Delta t}{2\Delta x^2} T_2^k - \frac{\alpha\Delta t}{2\Delta x^2} T_1^k + \frac{\Delta t}{2} [f(x_1, t_k) + f(x_1, t_{k+1})]$$

To simplify, we get

$$\left(1 + \frac{\alpha\Delta t}{2\Delta x^2} T_1^{k+1}\right) - \frac{\alpha\Delta t}{2\Delta x^2} T_2^{k+1} = \left(1 - \frac{\alpha\Delta t}{2\Delta x^2} T_1^k\right) + \frac{\alpha\Delta t}{2\Delta x^2} T_2^k + \frac{\Delta t}{2} [f(x_1, t_k) + f(x_1, t_{k+1})] \quad (\text{Eq. 1})$$

To put this in matrix form, we get

$$\begin{aligned} & \begin{bmatrix} 1 + \frac{\alpha\Delta t}{2\Delta x^2} & -\frac{\alpha\Delta t}{2\Delta x^2} \end{bmatrix}_{1 \times 2} \times \begin{bmatrix} T_1^{k+1} \\ T_2^{k+1} \end{bmatrix}_{2 \times 1} \\ &= \begin{bmatrix} 1 - \frac{\alpha\Delta t}{2\Delta x^2} & \frac{\alpha\Delta t}{2\Delta x^2} \end{bmatrix}_{1 \times 2} \times \begin{bmatrix} T_1^k \\ T_2^k \end{bmatrix}_{2 \times 1} + \frac{\Delta t}{2} [f(x_1, t_k) + f(x_1, t_{k+1})] \end{aligned}$$

Likewise, when  $i = 2$

$$\frac{T_2^{k+1} - T_2^k}{\Delta t} = \frac{\alpha}{2} \left[ \frac{T_3^k - 2T_2^k + T_1^k}{\Delta x^2} + \frac{T_3^{k+1} - 2T_2^{k+1} + T_1^{k+1}}{\Delta x^2} \right] + \frac{1}{2} [f(x_2, t_k) + f(x_2, t_{k+1})]$$

After changing some orders, we put  $T_i^{k+1}$  on the left-hand side of the equation and  $T_i^k$  on the right-hand side

$$\begin{aligned}
& -\frac{\alpha\Delta t}{2\Delta x^2}T_1^{k+1} + \left(1 + \frac{\alpha\Delta t}{\Delta x^2}T_2^{k+1}\right) - \frac{\alpha\Delta t}{2\Delta x^2}T_3^{k+1} \\
& = \frac{\alpha\Delta t}{2\Delta x^2}T_1^k + \left(1 - \frac{\alpha\Delta t}{\Delta x^2}\right)T_2^k + \frac{\alpha\Delta t}{2\Delta x^2}T_3^k + \frac{\Delta t}{2}[f(x_2, t_k) + f(x_2, t_{k+1})]
\end{aligned} \tag{Eq. 2}$$

In matrix form,

$$\begin{aligned}
& \begin{bmatrix} -\frac{\alpha\Delta t}{2\Delta x^2} & 1 + \frac{\alpha\Delta t}{\Delta x^2} & -\frac{\alpha\Delta t}{2\Delta x^2} \end{bmatrix}_{1 \times 3} \times \begin{bmatrix} T_1^{k+1} \\ T_2^{k+1} \\ T_3^{k+1} \end{bmatrix}_{3 \times 1} \\
& = \begin{bmatrix} \frac{\alpha\Delta t}{2\Delta x^2} & 1 - \frac{\alpha\Delta t}{\Delta x^2} & \frac{\alpha\Delta t}{2\Delta x^2} \end{bmatrix}_{1 \times 3} \times \begin{bmatrix} T_1^k \\ T_2^k \\ T_3^k \end{bmatrix}_{3 \times 1} + \frac{\Delta t}{2}[f(x_2, t_k) + f(x_2, t_{k+1})]
\end{aligned}$$

When  $i = n - 1$

$$\frac{T_{n-1}^{k+1} - T_{n-1}^k}{\Delta t} = \frac{\alpha}{2} \left[ \frac{T_n^k - 2T_{n-1}^k + T_{n-2}^k}{\Delta x^2} + \frac{T_n^{k+1} - 2T_{n-1}^{k+1} + T_{n-2}^{k+1}}{\Delta x^2} \right] + \frac{1}{2}[f(x_{n-1}, t_k) + f(x_{n-1}, t_{k+1})]$$

Once again, we put  $T_i^{k+1}$  on the left-hand side of the equation and  $T_i^k$  on the right hand side

$$\begin{aligned}
& -\frac{\alpha\Delta t}{2\Delta x^2}T_{n-2}^{k+1} + \left(1 + \frac{\alpha\Delta t}{\Delta x^2}T_{n-1}^{k+1}\right) - \frac{\alpha\Delta t}{2\Delta x^2}T_n^{k+1} \\
& = \frac{\alpha\Delta t}{2\Delta x^2}T_{n-2}^k + \left(1 - \frac{\alpha\Delta t}{\Delta x^2}\right)T_{n-1}^k + \frac{\alpha\Delta t}{2\Delta x^2}T_n^k + \frac{\Delta t}{2}[f(x_{n-1}, t_k) + f(x_{n-1}, t_{k+1})]
\end{aligned} \tag{Eq. n-1}$$

In matrix form

$$\begin{aligned}
& \begin{bmatrix} -\frac{\alpha\Delta t}{2\Delta x^2} & 1 + \frac{\alpha\Delta t}{\Delta x^2} & -\frac{\alpha\Delta t}{2\Delta x^2} \end{bmatrix}_{1 \times 3} \times \begin{bmatrix} T_{n-2}^{k+1} \\ T_{n-1}^{k+1} \\ T_n^{k+1} \end{bmatrix}_{3 \times 1} \\
& = \begin{bmatrix} \frac{\alpha\Delta t}{2\Delta x^2} & 1 - \frac{\alpha\Delta t}{\Delta x^2} & \frac{\alpha\Delta t}{2\Delta x^2} \end{bmatrix}_{1 \times 3} \times \begin{bmatrix} T_{n-2}^k \\ T_{n-1}^k \\ T_n^k \end{bmatrix}_{3 \times 1} + \frac{\Delta t}{2}[f(x_{n-1}, t_k) + f(x_{n-1}, t_{k+1})]
\end{aligned}$$

Finally, when  $i = n$

$$\frac{T_n^{k+1} - T_n^k}{\Delta t} = \frac{\alpha}{2} \left[ \frac{\mathbf{T}_{n+1}^k - 2T_n^k + T_{n-1}^k}{\Delta x^2} + \frac{\mathbf{T}_{n+1}^{k+1} - 2T_n^{k+1} + T_{n-1}^{k+1}}{\Delta x^2} \right] + \frac{1}{2} [f(x_n, t_k) + f(x_n, t_{k+1})]$$

Here,  $\mathbf{T}_{n+1}^k = T_n^k$  and  $\mathbf{T}_{n+1}^{k+1} = T_n^{k+1}$  because of the BCs.

Putting  $T_i^{k+1}$  on the left hand side of the equation and  $T_i^k$  on the right-hand side leads to

$$-\frac{\alpha \Delta t}{2 \Delta x^2} T_{n-1}^{k+1} + \left(1 + \frac{\alpha \Delta t}{2 \Delta x^2}\right) T_n^{k+1} = \frac{\alpha \Delta t}{2 \Delta x^2} T_{n-1}^k + \left(1 - \frac{\alpha \Delta t}{2 \Delta x^2}\right) T_n^k + \frac{\Delta t}{2} [f(x_n, t_k) + f(x_n, t_{k+1})] \quad (\text{Eq. } n)$$

In matrix form,

$$\begin{bmatrix} -\frac{\alpha \Delta t}{2 \Delta x^2} & 1 + \frac{\alpha \Delta t}{2 \Delta x^2} \end{bmatrix}_{1 \times 2} \times \begin{bmatrix} T_{n-1}^{k+1} \\ T_n^{k+1} \end{bmatrix}_{2 \times 1} = \begin{bmatrix} \frac{\alpha \Delta t}{2 \Delta x^2} & 1 - \frac{\alpha \Delta t}{2 \Delta x^2} \end{bmatrix}_{1 \times 2} \times \begin{bmatrix} T_{n-1}^k \\ T_n^k \end{bmatrix}_{2 \times 1} + \frac{\Delta t}{2} [f(x_n, t_k) + f(x_n, t_{k+1})]$$

Putting all these  $n$  linear equations [Eq. 1](#), [Eq. 2](#), ... [Eq. n-1](#), [Eq. n](#) together in matrix form, we arrive at

$$A_{n \times n} \vec{T}_{n \times 1}^{k+1} = B_{n \times n} \vec{T}_{n \times 1}^k + \vec{f}_{n \times 1} = \vec{b}_{n \times 1}$$

where

$$\vec{T}^{k+1} = \begin{bmatrix} T_1^{k+1} \\ T_2^{k+1} \\ \vdots \\ \vdots \\ T_{n-1}^{k+1} \\ T_n^{k+1} \end{bmatrix}_{n \times 1} \quad \vec{T}^k = \begin{bmatrix} T_1^k \\ T_2^k \\ \vdots \\ \vdots \\ T_{n-1}^k \\ T_n^k \end{bmatrix}_{n \times 1} \quad \vec{f} = \frac{\Delta t}{2} \begin{bmatrix} f(x_1, t_k) + f(x_1, t_{k+1}) \\ f(x_2, t_k) + f(x_2, t_{k+1}) \\ \vdots \\ \vdots \\ f(x_{n-1}, t_k) + f(x_{n-1}, t_{k+1}) \\ f(x_n, t_k) + f(x_n, t_{k+1}) \end{bmatrix}_{n \times 1}$$

$$A = \begin{bmatrix} 1 + \frac{\alpha \Delta t}{2\Delta x^2} & -\frac{\alpha \Delta t}{2\Delta x^2} & 0 & \dots & \dots & 0 \\ -\frac{\alpha \Delta t}{2\Delta x^2} & 1 + \frac{\alpha \Delta t}{\Delta x^2} & -\frac{\alpha \Delta t}{2\Delta x^2} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \dots & 0 & -\frac{\alpha \Delta t}{2\Delta x^2} & 1 + \frac{\alpha \Delta t}{\Delta x^2} & -\frac{\alpha \Delta t}{2\Delta x^2} \\ 0 & \dots & 0 & 0 & -\frac{\alpha \Delta t}{2\Delta x^2} & 1 + \frac{\alpha \Delta t}{2\Delta x^2} \end{bmatrix}_{n \times n}$$

$$B = \begin{bmatrix} 1 - \frac{\alpha \Delta t}{2\Delta x^2} & \frac{\alpha \Delta t}{2\Delta x^2} & 0 & \dots & \dots & 0 \\ \frac{\alpha \Delta t}{2\Delta x^2} & 1 - \frac{\alpha \Delta t}{\Delta x^2} & \frac{\alpha \Delta t}{2\Delta x^2} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \dots & 0 & \frac{\alpha \Delta t}{2\Delta x^2} & 1 - \frac{\alpha \Delta t}{\Delta x^2} & \frac{\alpha \Delta t}{2\Delta x^2} \\ 0 & \dots & 0 & 0 & \frac{\alpha \Delta t}{2\Delta x^2} & 1 - \frac{\alpha \Delta t}{2\Delta x^2} \end{bmatrix}_{n \times n}$$

$$b = \begin{bmatrix} \left(1 - \frac{\alpha \Delta t}{2\Delta x^2}\right) T_1^k + \frac{\alpha \Delta t}{2\Delta x^2} T_2^k + \frac{\Delta t}{2} [f(x_1, t_k) + f(x_1, t_{k+1})] \\ \frac{\alpha \Delta t}{2\Delta x^2} T_1^k + \left(1 - \frac{\alpha \Delta t}{\Delta x^2}\right) T_2^k + \frac{\alpha \Delta t}{2\Delta x^2} T_3^k + \frac{\Delta t}{2} [f(x_2, t_k) + f(x_2, t_{k+1})] \\ \vdots \\ \frac{\alpha \Delta t}{2\Delta x^2} T_{n-1}^k + \left(1 - \frac{\alpha \Delta t}{\Delta x^2}\right) T_n^k + \frac{\alpha \Delta t}{2\Delta x^2} T_n^k + \frac{\Delta t}{2} [f(x_{n-1}, t_k) + f(x_{n-1}, t_{k+1})] \\ \frac{\alpha \Delta t}{2\Delta x^2} T_{n-1}^k + \left(1 - \frac{\alpha \Delta t}{2\Delta x^2}\right) T_n^k + \frac{\Delta t}{2} [f(x_n, t_k) + f(x_n, t_{k+1})] \end{bmatrix}_{n \times 1}$$

Note that  $A$  and  $B$  are triangular matrices. We can use the sparse matrix solver in MATLAB to create them and they are easily to be computed. In MATLAB, we simply use  $\vec{T}^{k+1} = A \setminus \vec{b}$  to find the temperature rise with respect to time.

Let us recall from Figure 2, the idea of a dithering laser beam heat source  $f(x, t)$  on a 1-D rod of Eq. 2.1 can be written differently:

$$\frac{\partial T}{\partial t} = \alpha_T \frac{\partial^2 T}{\partial x^2} + \frac{\alpha_T}{K_T} f(x, t), \quad 0 \leq x \leq L_x$$

$$f(x, t) = \frac{I_0}{\sqrt{2\pi d^2}} e^{-\frac{(x-x_c(t))^2}{2d^2}}, \quad x_c(t) = x_0 + a \cdot \sin\left(\frac{2\pi t}{period}\right), \quad x_0 = \frac{L_x}{2}$$
(Eq. 3.4)

Where  $d$  is the standard deviation of the Gaussian heat source. Figure 7 demonstrates what the final temperature rise looks like for time = 1 using the Crank-Nicolson method with number of points  $n = 200$ . All the input parameters are listed in Table 1. The MATLAB code is attached in Appendix A.

Input	Value	Unit
$\alpha_T$	1	m <sup>2</sup> /s
$K_T$	1	W/(m*k)
$I_0$	1	W/m <sup>3</sup>
$d$	0.02	m
$L_x$	1	m
$x_0$	0.5	m
$a$	0.25	m
$period$	1	s

Table 1. 1-D rod MATLAB input parameters for dithering laser beam

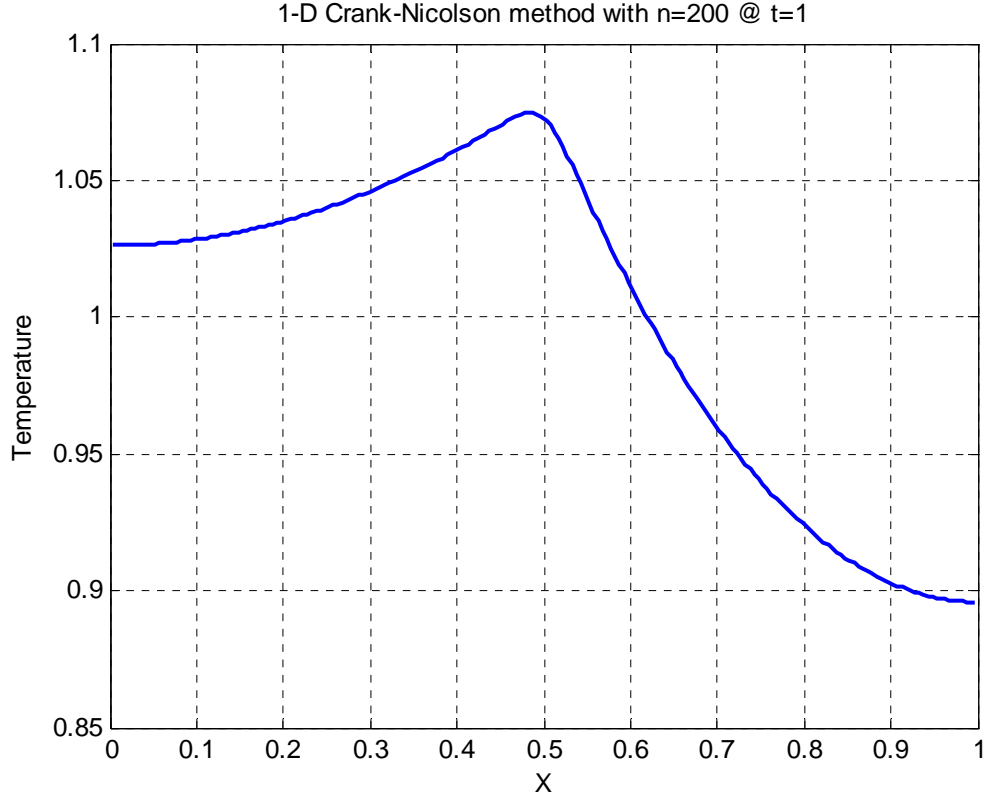


Figure 7. 1-D Crank-Nicolson method

## B. THE FAST FOURIER TRANSFORM (FFT) METHOD

Recall that in (Eq. 2.10) we have shown that the solution of the 1-D nonhomogeneous heat equation with insulating boundary conditions can be expressed as eigenfunction expansion, where the eigenfunctions are cosine functions.

Now we show how to use FFT to implement the eigenfunction expansion for the purpose of fast summation. This can be achieved in several steps.

First, we even extend the initial condition from  $[0,1]$  to  $[1,2]$  and then make it periodic with period 2. After that, we map the interval  $[0,2]$  to  $[0,2\pi]$ . By doing this, the Fourier series will only contain cosine terms.



Second, we apply the discrete Fourier transform to the heat equation  $\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + f(x, t)$ :

$$F\left[\frac{\partial u}{\partial t}\right] = F\left[\frac{\partial^2 u}{\partial x^2}\right] + F[f(x, t)]$$

where  $F$  denotes the Fourier transform operator.

Now we need to calculate Fourier transformations of derivatives of  $u(x, t)$ . We begin by recalling the spatial Fourier transform of  $u(x, t)$ :

$$F[u(x, t)] = \hat{u}(k, t) = \frac{1}{2\pi} \int_0^{2\pi} u(x, t) e^{ikx} dx$$

Note that this is also a function of time; it is an ordinary Fourier transform with time  $t$  fixed. To obtain a Fourier transform in space, we multiply  $e^{ikx}$  and integrate. Spatial Fourier transforms of time derivatives can be derived easily because the spatial Fourier transforms of a time derivative equals the time derivative of the Fourier transform:

$$F\left[\frac{\partial u(x, t)}{\partial t}\right] = \frac{1}{2\pi} \int_0^{2\pi} \frac{\partial u(x, t)}{\partial t} e^{ikx} dx = \frac{\partial}{\partial t} \left[ \frac{1}{2\pi} \int_0^{2\pi} u(x, t) e^{ikx} dx \right] = \frac{\partial}{\partial t} \hat{u}(k, t)$$

For spatial Fourier transform of spatial derivatives, the method of integrating by parts can be used:

$$F\left[\frac{\partial u(x, t)}{\partial x}\right] = \frac{1}{2\pi} \int_0^{2\pi} \frac{\partial u(x, t)}{\partial x} e^{ikx} dx = \frac{u(x, t) e^{ikx}}{2\pi} \Big|_{x=0}^{x=2\pi} - ik\pi \left[ \frac{1}{2\pi} \int_0^{2\pi} u(x, t) e^{ikx} dx \right] = -ik\pi \hat{u}(k, t)$$

Here, we assume  $u \rightarrow 0$  as  $x \rightarrow 0$  or  $2\pi$ , and then the first term of this equation vanishes.

In general, the Fourier transform of the  $n^{th}$  derivatives of a function with respect to  $x$  equals  $(-ik\pi)^n$  times the Fourier transform of the function, assuming that  $u(x,t) \rightarrow 0$  sufficiently fast as  $x$  approaches periodic endpoint  $x \rightarrow 0$  and  $x \rightarrow 2\pi$ .

Likewise, Fourier transforms of a second derivative can be obtained:

$$F\left[\frac{\partial^2 u(x,t)}{\partial^2 x}\right] = -ik\pi F\left[\frac{\partial u(x,t)}{\partial x}\right] = (-ik\pi)^2 \hat{u}(k,t) = -k^2 \pi^2 \hat{u}(k,t)$$

Finally, we can conclude that  $\frac{d\hat{u}}{dt} = (-ik\pi)^2 \hat{u} + \hat{f}(k,t) = -k^2 \pi^2 \hat{u} + \hat{f}(k,t)$ . The factor  $\pi$  here is due to the fact that we map the interval  $[0,2]$  to  $[0,2\pi]$ . Using the method of integrating factor and multiplying  $e^{k^2 \pi^2 t}$  on this equation, we find

$$e^{k^2 \pi^2 t} \frac{d\hat{u}}{dt} = -k^2 \pi^2 e^{k^2 \pi^2 t} \hat{u} + e^{k^2 \pi^2 t} \hat{f}(k,t)$$

We rewrite this ODE as

$$\frac{d}{dt} \left( \hat{u} e^{k^2 \pi^2 t} \right) = e^{k^2 \pi^2 t} \hat{f}(k,t)$$

Integrating both sides yields

$$\hat{u}(k,t) = \hat{u}(k,0) e^{-k^2 \pi^2 t} + e^{-k^2 \pi^2 t} \int_0^t e^{k^2 \pi^2 s} \hat{f}(k,s) ds = \hat{u}(k,0) e^{-k^2 \pi^2 t} + \int_0^t e^{k^2 \pi^2 (s-t)} \hat{f}(k,s) ds$$

Here, we make the change of variables  $\tau = t - s$  to conduct our following derivation.

This equation becomes

$$\hat{u}(k,t) = \hat{u}(k,0) e^{-k^2 \pi^2 t} + \int_0^t e^{-k^2 \pi^2 \tau} \hat{f}(k,t-\tau) d\tau$$

In order to estimate this integral, we approximate  $\hat{f}(k,t-\tau)$  by a linear function:

$$\hat{f}(k,t-\tau) \approx \hat{f}(k,t) + \frac{\hat{f}(k,0) - \hat{f}(k,t)}{t} \tau$$

Then, it follows that

$$\begin{aligned}
\hat{u}(k, t) &= \hat{u}(k, 0)e^{-k^2\pi^2 t} + \int_0^t e^{-k^2\pi^2\tau} \hat{f}(k, t-\tau) d\tau \\
&\approx \hat{u}(k, 0)e^{-k^2\pi^2 t} + \int_0^t e^{-k^2\pi^2\tau} \left[ \hat{f}(k, t) + \frac{\hat{f}(k, 0) - \hat{f}(k, t)}{t} \tau \right] d\tau \\
&= \hat{u}(k, 0)e^{-k^2\pi^2 t} + \hat{f}(k, t) \frac{1}{k^2\pi^2} (1 - e^{-k^2\pi^2 t}) + \frac{\hat{f}(k, 0) - \hat{f}(k, t)}{t} \int_0^t e^{-k^2\pi^2\tau} \tau d\tau \\
&= \hat{u}(k, 0)e^{-k^2\pi^2 t} + \hat{f}(k, t) \frac{1}{k^2\pi^2} (1 - e^{-k^2\pi^2 t}) \\
&\quad + \frac{\hat{f}(k, 0) - \hat{f}(k, t)}{t} \left[ -\frac{t}{k^2\pi^2} e^{-k^2\pi^2 t} + \frac{1}{k^4\pi^4} (1 - e^{-k^2\pi^2 t}) \right] \\
&= \hat{u}(k, 0)e^{-k^2\pi^2 t} + \hat{f}(k, t) \frac{1}{k^2\pi^2} (1 - e^{-k^2\pi^2 t}) \\
&\quad + \frac{\hat{f}(k, 0) - \hat{f}(k, t)}{t} \underbrace{\left[ \frac{1}{k^4\pi^4} - \frac{1 + tk^2\pi^2}{k^4\pi^4} e^{-k^2\pi^2 t} \right]}_{\frac{1 - (1 + tk^2\pi^2)e^{-k^2\pi^2 t}}{k^4\pi^4}}
\end{aligned}$$

Once  $\hat{u}(k, t)$  is found, apply the inverse discrete Fourier transform to it to get  $u(x, t)$ .

In our MATAB code, as attached in Appendix B,  $t$  is replaced by  $dt$  because in each step along the time direction we march  $dt$ . The first component in  $h_1$  or  $h_2$  corresponds to the case where  $k = 0$ . We apply H'Lospital's rule to find the value when  $k = 0$ . More specifically, we have used the formulas:

$$\begin{aligned}
\lim_{k \rightarrow 0} \frac{1}{k^2\pi^2} (1 - e^{-k^2\pi^2 t}) &\stackrel{\lambda=k^2\pi^2}{=} \lim_{\lambda \rightarrow 0} \frac{1 - e^{-\lambda t}}{\lambda} \left( \frac{0}{0} \text{ type} \right) = \lim_{\lambda \rightarrow 0} \frac{-(-t)e^{-\lambda t}}{1} = t \\
\lim_{k \rightarrow 0} \frac{1 - (1 + tk^2\pi^2)e^{-k^2\pi^2 t}}{k^4\pi^4} &\stackrel{\lambda=k^2\pi^2}{=} \lim_{\lambda \rightarrow 0} \frac{1 - (1 + t\lambda)e^{-\lambda t}}{\lambda^2} \left( \frac{0}{0} \text{ type} \right) = \lim_{\lambda \rightarrow 0} \frac{-te^{-\lambda t} + t(1 + t\lambda)e^{-\lambda t}}{2\lambda} \left( \frac{0}{0} \text{ type} \right)
\end{aligned}$$

$$\begin{aligned}
&= \frac{1}{2} \lim_{\lambda \rightarrow 0} \left\{ t^2 e^{-\lambda t} + t \left[ t e^{-\lambda t} - t(1 + t\lambda) e^{-\lambda t} \right] \right\} \\
&= \frac{1}{2} \lim_{\lambda \rightarrow 0} \left\{ t^2 e^{-\lambda t} + t(-\lambda t^2) \right\} \\
&= \frac{1}{2} t^2
\end{aligned}$$

Figure 8 shows the result using FFT method in MATLAB where all the input parameters are from Table 1.

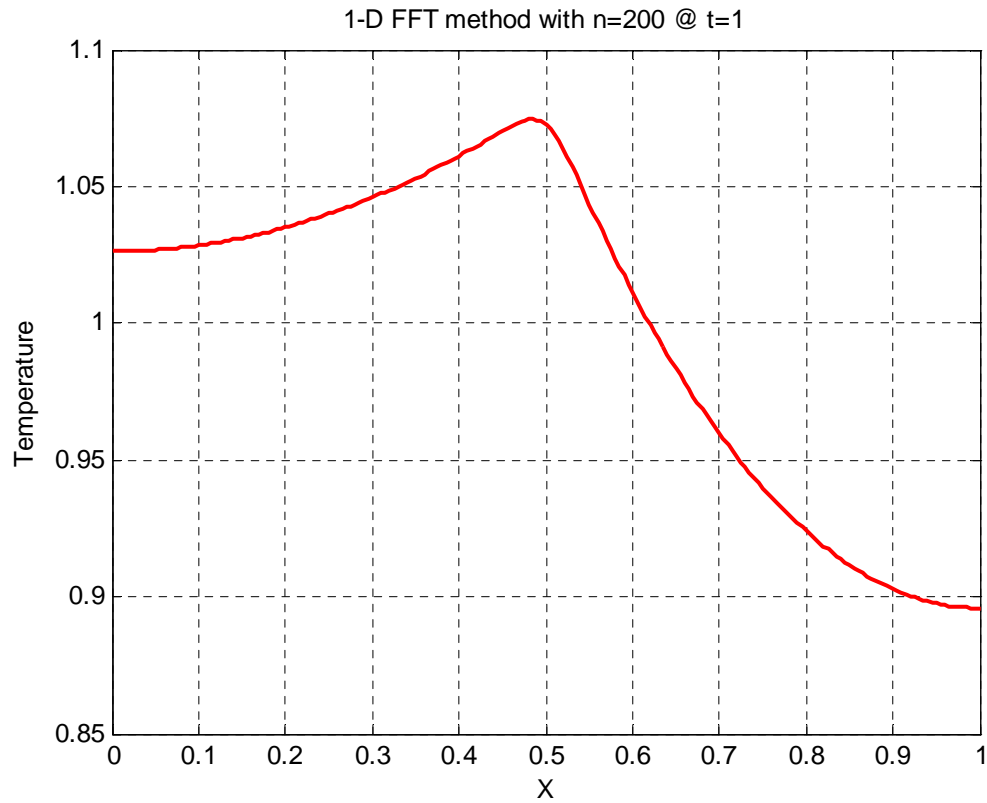


Figure 8. 1-D FFT method

### C. COMSOL MULTIPHYSICS 4.0A

COMSOL is one of the popular computer simulation software programs used to model and simply translate real-world physical laws into the real world in the virtual form using the finite element method. COMSOL is a commercial problem-solving tool that produces results quickly. However, it is more important to investigate the accuracy of the numerical results. We compare the solution differences between MATALB and COMSOL in section D.

Figure 9 is the result from COMSOL and there is no surprise that both results from MATLAB and COMSOL are very close. The step-by-step process of creating this 1-D dithering laser problem using (Eq. 3.4) in COMSOL is attached in Appendix C.

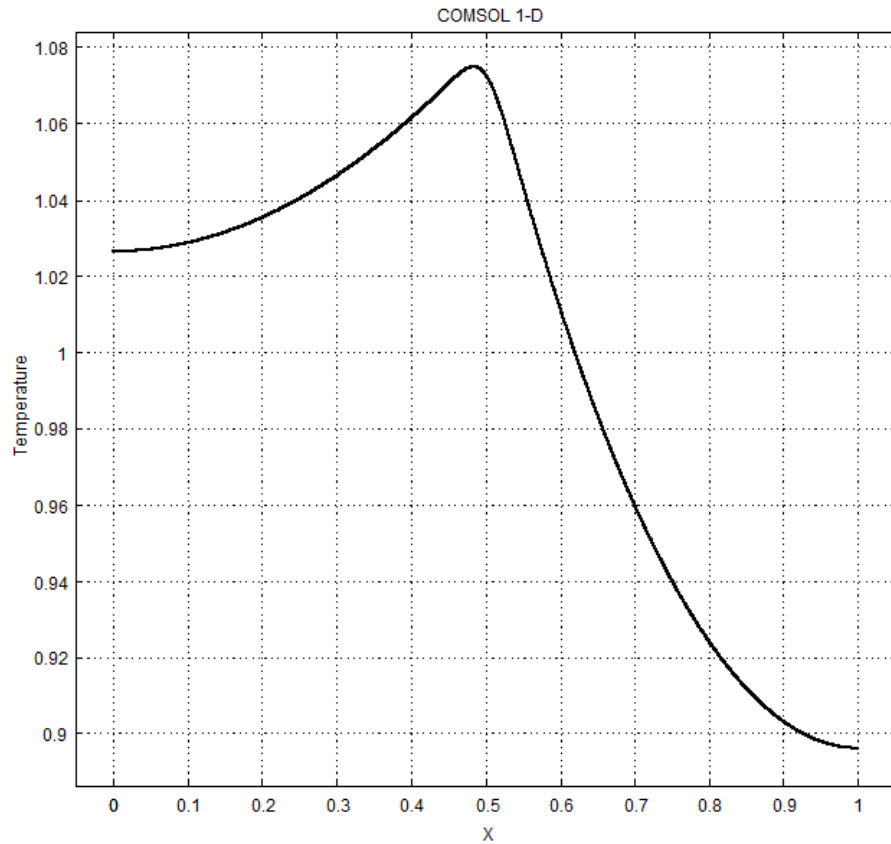


Figure 9. 1-D COMSOL

#### D. COMPARISON ON A MODEL PROBLEM

We compare the results from the 1-D Crank-Nicolson method and COMSOL and plot the relative difference in Figure 10, where the number of points is increased from 200 to 2001. The relative error between MATLAB and COMSOL results is about  $10^{-4}$ , which is really small and tolerable. Here, we only show the comparison between Crank-Nicolson method in MATLAB and COMSOL, and the comparison in FFT method and COMSOL are also pretty similar. After several tryouts, it is intuitive to see that the relative error goes down once we increase the number of points  $N$ , but it takes longer to do so.

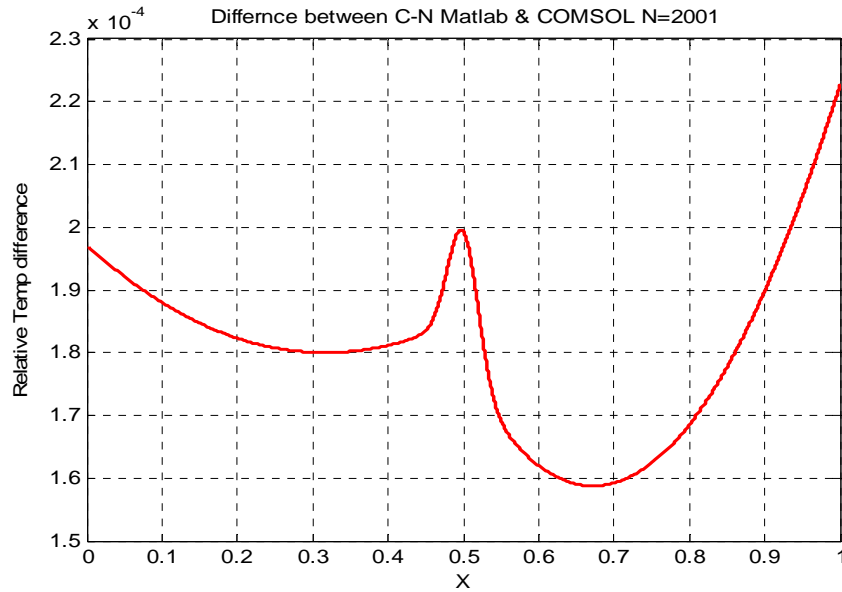


Figure 10. Relative error plot in 1-D Crank-Nicolson and COMSOL

#### E. REAL PROBLEM SIMULATION (STEEL AISI 4340)

After we have demonstrated that both MATLAB and COMSOL are giving us acceptable results, we can choose a specific material to simulate a real problem. The material we use in our 1-D plot comparison is Steel AISI 4340, a built-in material in COMSOL which has the material contents listed in Table 2:

Property	Name	Value	Unit
Heat Capacity	$C_p$	475	J/(kg*K)
Density	$\rho$	7850	Kg/(m <sup>3</sup> )
Thermal Conductivity	$K_T$	44.5	W/(m*K)
Thermal Diffusivity	$\alpha_T$	$\frac{K_T}{\rho \cdot C_p}$	m <sup>2</sup> /s
Melting Point		1783	K

Table 2. Thermal property of Steel AISI 4340

The dithering laser we deploy has the inputs in Table 3:

Laser Input	Name	Value	Unit
Magnitude of Gaussian source	$I_0$	1.0e9	W/(m <sup>3</sup> )
Effective radius of Gaussian heat source	$d$	0.02	m
Rotating frequency	$Freq$	1	Hz
Laser stop time	$Time$	1	s
Center of rotation	$x_0$	0.5	m
Rotating radius	$a$	0.25	m

Table 3. 1-D dithering laser input on Steel AISI 4340

Figure 11 from MATLAB and Figure 12 from COMSOL depict the results of temperature rise using the contents from Table 2 and Table 3. Both results are very close. Figure 12 is from the Crank-Nicolson method and the FFT method yields a very similar result. It is clear that the maximum temperature rise decreases as the rotating period decreases; in other words, the higher the frequency is, the less the temperature rise increases. At  $t = 1s$ , Figure 13 shows the maximum temperature rise of the steel AISI

4340 versus the frequency (reciprocal of the period) of the rotating Gaussian beam from 1 Hz to 100 Hz . It is observed that a higher frequency leads to lower the maximum temperature rise. It should be pointed out that our results are consistent with earlier analytical studies for the semi-infinite domain [5].

Figure 14 shows the temperature rise as a function of time at a fixed point  $x = 0.75$  of the material with dithering laser beam period=0.1 sec. The temperature increases as the dithering laser beam moves closer to the point and the temperature almost stays steady as the laser beam moves away. The overall line shape behaves as an increasing function of time.

Figure 15 shows the maximum temperature rise versus frequency with two different heat sources  $I_0$ . The higher  $I_0$ , the higher the temperature rise.

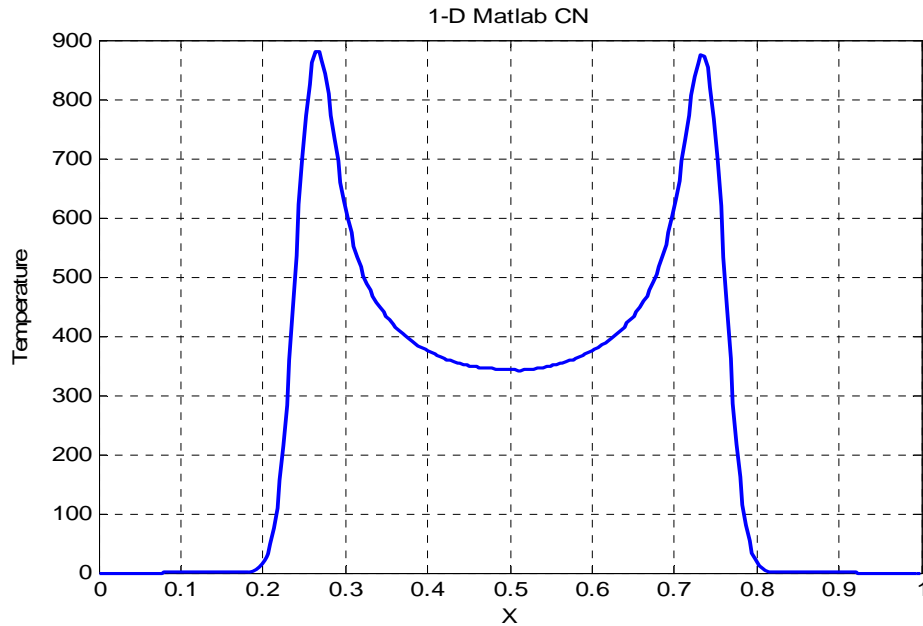


Figure 11. Temperature rise on the material of steel AISI 4340 using the 1-D Crank-Nicholson method



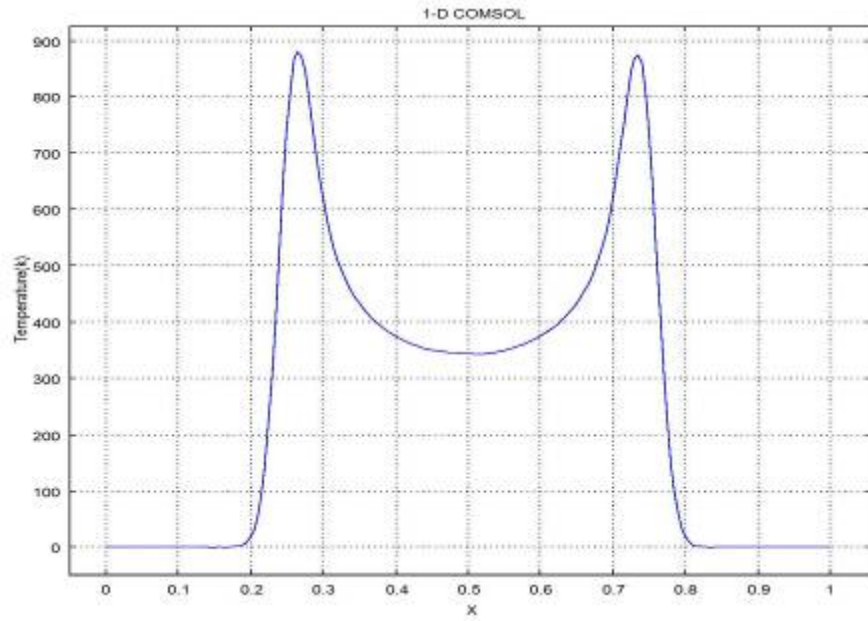


Figure 12. Temperature rise on the material of steel AISI 4340 using 1-D COMSOL

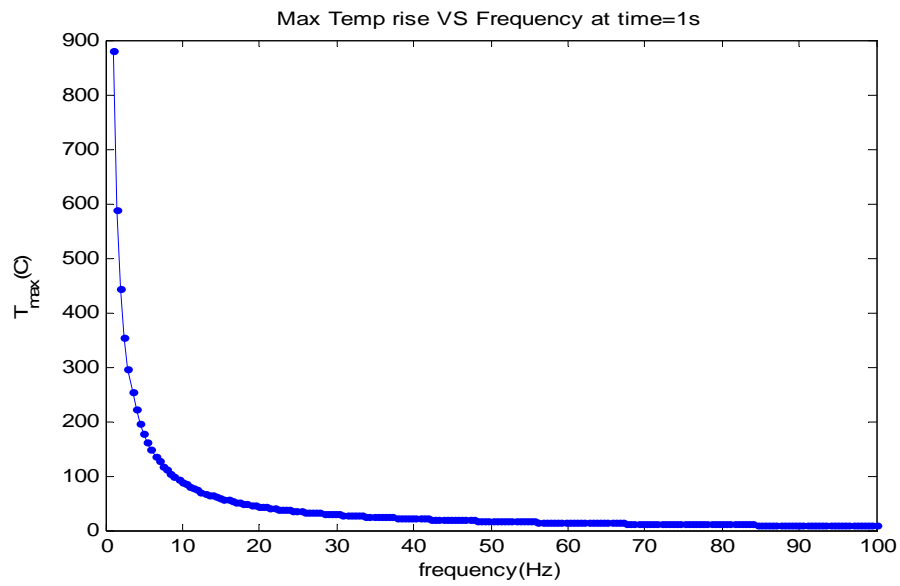


Figure 13. 1-D maximum temperature rise of steel AISI 4340 versus the frequency of the dithering laser beam

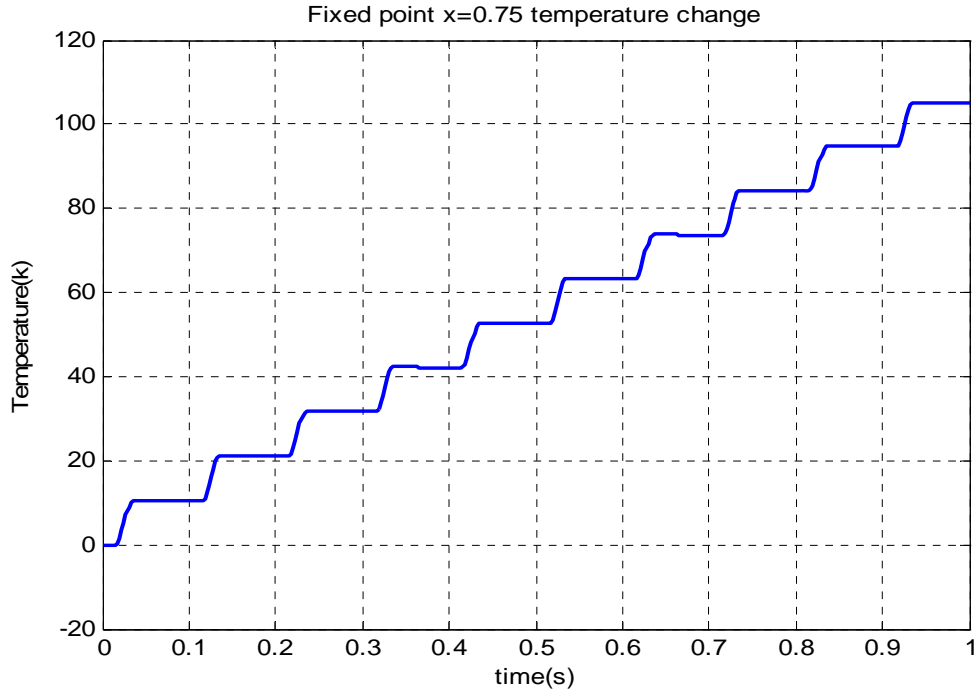


Figure 14. 1-D temperature change with time at a fixed point  $x=0.75$  of steel AISI 4340 with dithering laser beam period=0.1s

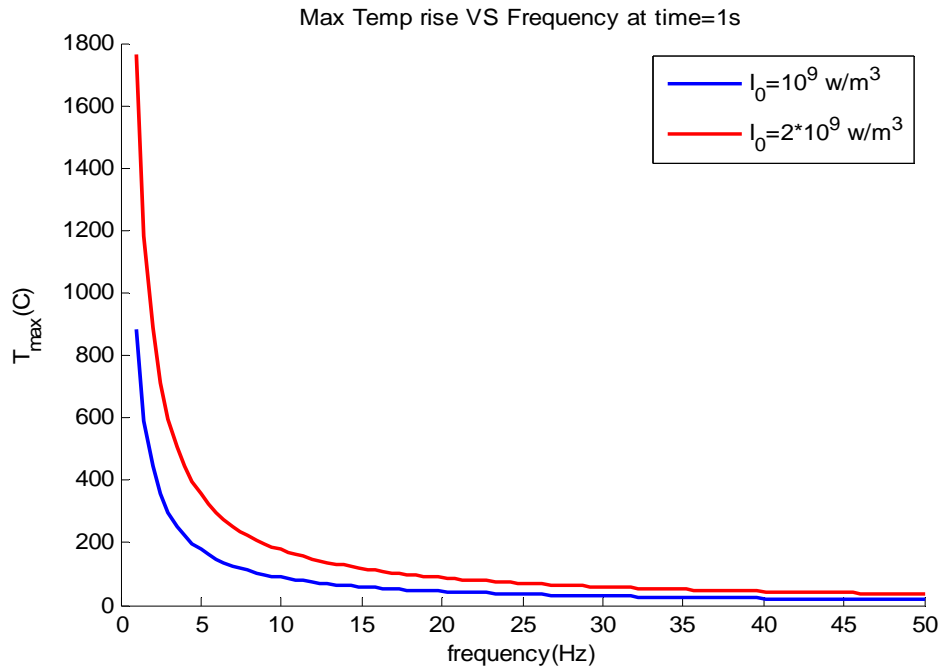


Figure 15. The maximum temperature rise of steel AISI 4340 versus frequency with two different heat source  $I_0$

## IV. NUMERICAL SOLUTION FOR A TRANSIENT, TWO-DIMENSIONAL TEMPERATURE DISTRIBUTION IN A FINITE FILM DUE TO A ROTATING OR DITHERING LASER BEAM

Now we extend our work in 1-D to 2-D.

### A. THE CRANK-NICOLSON METHOD

We start with the 2-D nonhomogeneous heat equation

$$\frac{\partial u}{\partial t} = \alpha_T \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + \frac{\alpha_T}{K_T} f(x, y, t), \quad 0 \leq x \leq L_x, \quad 0 \leq y \leq L_y$$

$$f(x, y, t) = \frac{I_0}{2\pi d^2} e^{\frac{-((x-x_c(t))^2 + (y-y_c(t))^2)}{2d^2}},$$

$$x_c(t) = x_0 + a \cdot \cos\left(\frac{2\pi t}{\text{period}}\right), \quad x_0 = \frac{L_x}{2}$$

$$y_c(t) = y_0 + b \cdot \sin\left(\frac{2\pi t}{\text{period}}\right), \quad y_0 = \frac{L_y}{2}$$
(Eq. 4.1)

(Eq. 4.1) can be illustrated as in Figure 16. The heat source  $f(x, y, t)$  created by the laser beam is illustrated in Figure 17.

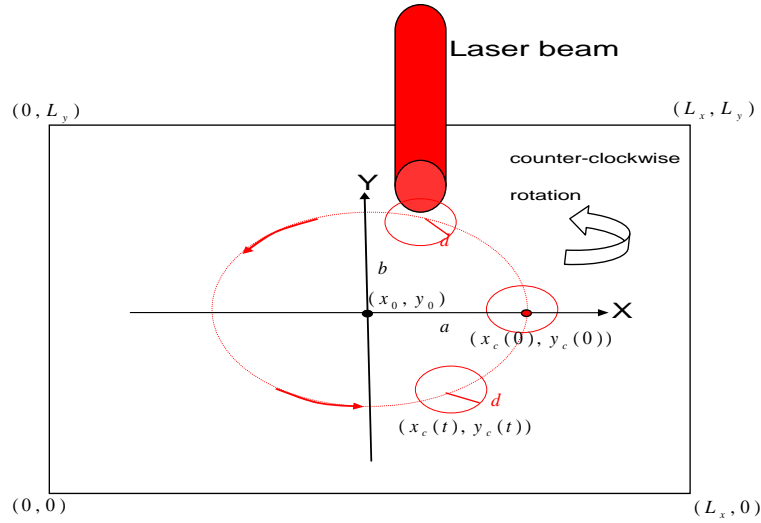
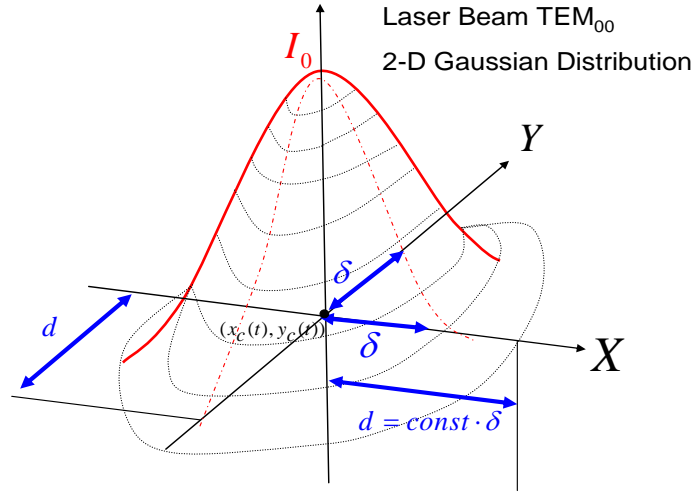


Figure 16. 2-D schematic of the laser beam and the finite work piece



$$\text{Heat source: } f(x, y, t) = \frac{I_0}{2\pi d^2} e^{-\frac{((x-x_c(t))^2 + (y-y_c(t))^2)}{2d^2}}$$

Figure 17. Laser beam creates heat source as a Gaussian distribution [After 2].

We discretize this 2-D rectangular domain as shown in Figure 18. In order to facilitate our presentation, we set  $L_x = L_y = 1$  and the thermal properties  $\alpha_T$  and  $K_T$  equal

1, thus the heat equation is simplified to the form  $\frac{\partial u}{\partial t} = \Delta u + f(x, y, t)$ .

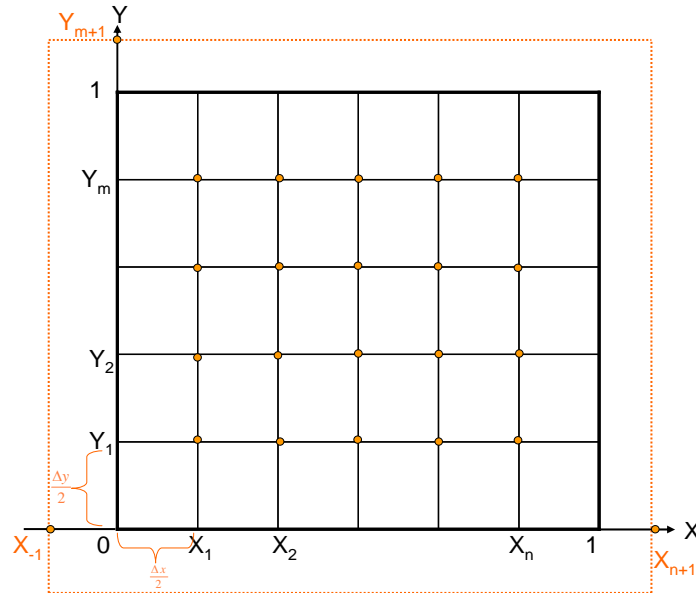


Figure 18. Discretization of a 2-D rectangular region

The discretized point is denoted as  $(x_i, y_j)$ , where  $i$  goes from 1 to  $n$  and  $j$  varies from 1 to  $m$ . Here,  $n$  and  $m$  are the total number of grid points in  $x$  and  $y$  respectively. More explicitly,

$$x_i = \left(i - \frac{1}{2}\right) \Delta x, \quad \Delta x = \frac{(L_x - 0)}{n}, \quad i = 1, 2, \dots, n$$

$$y_j = \left(j - \frac{1}{2}\right) \Delta y, \quad \Delta y = \frac{(L_y - 0)}{m}, \quad j = 1, 2, \dots, m$$

The boundary conditions assume that the solid is insulated at the edges (boundaries) and the initial condition is that there is no temperature rise with respect to the ambient temperature initially. Thus, we have

$$\text{BCs: } \begin{aligned} \left. \frac{\partial u}{\partial x} \right|_{x=0} &= \left. \frac{\partial u}{\partial x} \right|_{x=1} = 0 \\ \left. \frac{\partial u}{\partial y} \right|_{y=0} &= \left. \frac{\partial u}{\partial y} \right|_{y=1} = 0 \end{aligned}$$

$$\text{ICs: } u(x, y, 0) = 0$$

We introduce the shorthand notation  $u_{i,j}^k \equiv u(x_i, y_j, t_k)$ . Then the Crank-Nicolson 2-D scheme of (Eq. 4.1) becomes

$$\begin{aligned} \frac{u_{i,j}^{k+1} - u_{i,j}^k}{\Delta t} &= \frac{1}{2} \left[ \frac{u_{i+1,j}^k - 2u_{i,j}^k + u_{i-1,j}^k}{\Delta x^2} + \frac{u_{i+1,j}^{k+1} - 2u_{i,j}^{k+1} + u_{i-1,j}^{k+1}}{\Delta x^2} \right] \\ &+ \frac{1}{2} \left[ \frac{u_{i,j+1}^k - 2u_{i,j}^k + u_{i,j-1}^k}{\Delta y^2} + \frac{u_{i,j+1}^{k+1} - 2u_{i,j}^{k+1} + u_{i,j-1}^{k+1}}{\Delta y^2} \right] \\ &+ \frac{1}{2} \left[ f(x_i, y_j, t_k) + f(x_i, y_j, t_{k+1}) \right] \end{aligned} \quad \begin{aligned} &1 \leq i \leq n, \quad 1 \leq j \leq m \\ &k = 0, 1, 2, \dots, \quad t_k = k \Delta t \end{aligned} \quad (\text{Eq. 4.2})$$

The BC can be satisfied by setting

$$\begin{aligned}
u_{1,j}^k &= u_{0,j}^k \\
u_{n,j}^k &= u_{n+1,j}^k \\
u_{i,1}^k &= u_{i,0}^k \\
u_{i,m}^k &= u_{i,m+1}^k
\end{aligned}$$

We can rearrange the temperature at the interior grid points to form a vector  $\vec{u}^k$  as depicted in Figure 19:

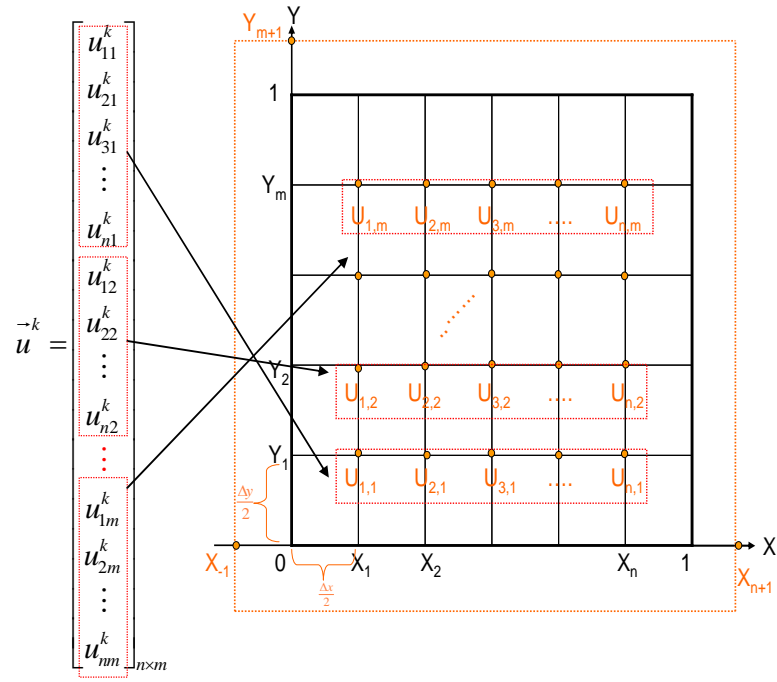


Figure 19. Putting the temperature of each point in a vector form  $\vec{u}^k$  after discretization

We will convert the linear system (Eq. 4.2) into the following matrix equation:

$$\begin{aligned}
A \vec{u}^{k+1} &= B \vec{u}^k + \vec{f} = \vec{b} \\
\vec{u}^{k=0} &= \vec{0}
\end{aligned}$$

where  $A$  and  $B$  are the matrices to be found,  $\left\{ \overset{\rightarrow}{u}^{k+1} \right\}$  are unknown and  $\left\{ \overset{\leftarrow}{u}^k \right\}$  are known.

We rewrite (Eq. 4.2) as

$$\begin{aligned} \overset{\rightarrow}{u}_{i,j}^{k+1} &= \overset{\leftarrow}{u}_{i,j}^k + \frac{\Delta t}{2\Delta x^2} [\overset{\leftarrow}{u}_{i+1,j}^k - 2\overset{\leftarrow}{u}_{i,j}^k + \overset{\leftarrow}{u}_{i-1,j}^k] + \frac{\Delta t}{2\Delta y^2} [\overset{\leftarrow}{u}_{i,j+1}^k - 2\overset{\leftarrow}{u}_{i,j}^k + \overset{\leftarrow}{u}_{i,j-1}^k] \\ &\quad + \frac{\Delta t}{2\Delta x^2} [\overset{\rightarrow}{u}_{i+1,j}^{k+1} - 2\overset{\rightarrow}{u}_{i,j}^{k+1} + \overset{\rightarrow}{u}_{i-1,j}^{k+1}] + \frac{\Delta t}{2\Delta y^2} [\overset{\rightarrow}{u}_{i,j+1}^{k+1} - 2\overset{\rightarrow}{u}_{i,j}^{k+1} + \overset{\rightarrow}{u}_{i,j-1}^{k+1}] \\ &\quad + \frac{\Delta t}{2} [f(x_i, y_j, t_k) + f(x_i, y_j, t_{k+1})] \end{aligned}$$

Putting  $\overset{\rightarrow}{u}^{k+1}$  on the left-hand side of the equation,  $\overset{\leftarrow}{u}^k$  and  $f$  on the right-hand side, we obtain:

$$\begin{aligned} &-\frac{\Delta t}{2\Delta x^2} \overset{\rightarrow}{u}_{i+1,j}^{k+1} + \left[ 1 + \frac{\Delta t}{\Delta x^2} + \frac{\Delta t}{\Delta y^2} \right] \overset{\rightarrow}{u}_{i,j}^{k+1} - \frac{\Delta t}{2\Delta x^2} \overset{\rightarrow}{u}_{i-1,j}^{k+1} - \frac{\Delta t}{2\Delta y^2} \overset{\rightarrow}{u}_{i,j+1}^{k+1} - \frac{\Delta t}{2\Delta y^2} \overset{\rightarrow}{u}_{i,j-1}^{k+1} \\ &= \frac{\Delta t}{2\Delta x^2} \overset{\leftarrow}{u}_{i+1,j}^k + \left[ 1 - \frac{\Delta t}{\Delta x^2} - \frac{\Delta t}{\Delta y^2} \right] \overset{\leftarrow}{u}_{i,j}^k + \frac{\Delta t}{2\Delta x^2} \overset{\leftarrow}{u}_{i-1,j}^k + \frac{\Delta t}{2\Delta y^2} \overset{\leftarrow}{u}_{i,j+1}^k + \frac{\Delta t}{2\Delta y^2} \overset{\leftarrow}{u}_{i,j-1}^k \quad (\text{Eq. 4.3}) \\ &\quad + \frac{\Delta t}{2} [f(x_i, y_j, t_k) + f(x_i, y_j, t_{k+1})] \end{aligned}$$

We start a simple case with  $i = 1, j = 1$  first,

$$\begin{aligned} &-\frac{\Delta t}{2\Delta x^2} \overset{\rightarrow}{u}_{2,1}^{k+1} + \left[ 1 + \frac{\Delta t}{\Delta x^2} + \frac{\Delta t}{\Delta y^2} \right] \overset{\rightarrow}{u}_{1,1}^{k+1} - \frac{\Delta t}{2\Delta x^2} \underbrace{\overset{\rightarrow}{u}_{0,1}^{k+1}}_{=\overset{\rightarrow}{u}_{1,1}^{k+1}} - \frac{\Delta t}{2\Delta y^2} \overset{\rightarrow}{u}_{1,2}^{k+1} - \frac{\Delta t}{2\Delta y^2} \underbrace{\overset{\rightarrow}{u}_{1,0}^{k+1}}_{=\overset{\rightarrow}{u}_{1,1}^{k+1}} \\ &= \frac{\Delta t}{2\Delta x^2} \overset{\leftarrow}{u}_{2,1}^k + \left[ 1 - \frac{\Delta t}{\Delta x^2} - \frac{\Delta t}{\Delta y^2} \right] \overset{\leftarrow}{u}_{1,1}^k + \frac{\Delta t}{2\Delta x^2} \underbrace{\overset{\leftarrow}{u}_{0,1}^k}_{=\overset{\leftarrow}{u}_{1,1}^k} + \frac{\Delta t}{2\Delta y^2} \overset{\leftarrow}{u}_{1,2}^k + \frac{\Delta t}{2\Delta y^2} \underbrace{\overset{\leftarrow}{u}_{1,0}^k}_{=\overset{\leftarrow}{u}_{1,1}^k} \\ &\quad + \frac{\Delta t}{2} [f(x_1, y_1, t_k) + f(x_1, y_1, t_{k+1})] \end{aligned}$$

Using the boundary conditions ( $\overset{\rightarrow}{u}_{0,1}^{k+1} = \overset{\rightarrow}{u}_{1,1}^{k+1}$ ,  $\overset{\rightarrow}{u}_{1,0}^{k+1} = \overset{\rightarrow}{u}_{1,1}^{k+1}$  and so on), we can simplify this equation

$$\begin{aligned}
& \left[ 1 + \frac{\Delta t}{2\Delta x^2} + \frac{\Delta t}{2\Delta y^2} \right] \mathbf{u}_{1,1}^{k+1} - \frac{\Delta t}{2\Delta x^2} \mathbf{u}_{2,1}^{k+1} - \frac{\Delta t}{2\Delta y^2} \mathbf{u}_{1,2}^{k+1} \\
& = \left[ 1 - \frac{\Delta t}{2\Delta x^2} - \frac{\Delta t}{2\Delta y^2} \right] \mathbf{u}_{1,1}^k + \frac{\Delta t}{2\Delta x^2} \mathbf{u}_{2,1}^k + \frac{\Delta t}{2\Delta y^2} \mathbf{u}_{1,2}^k + \frac{\Delta t}{2} [f(x_1, y_1, t_k) + f(x_1, y_1, t_{k+1})]
\end{aligned}$$

Recall the matrix form  $A\vec{u}^{k+1} = B\vec{u}^k + \vec{f}$ . We can obtain the corresponding components of  $A$  and  $B$  from this equation:

$$\begin{aligned}
A(1,1) &= 1 + \frac{\Delta t}{2\Delta x^2} + \frac{\Delta t}{2\Delta y^2} \\
A(1,2) &= -\frac{\Delta t}{2\Delta x^2} \\
A(1,n+1) &= -\frac{\Delta t}{2\Delta y^2} \\
B(1,1) &= 1 - \frac{\Delta t}{2\Delta x^2} - \frac{\Delta t}{2\Delta y^2} \\
B(1,2) &= \frac{\Delta t}{2\Delta x^2} \\
B(1,n+1) &= \frac{\Delta t}{2\Delta y^2}
\end{aligned}$$

and we will put  $\frac{\Delta t}{2} [f(x_1, y_1, t_k) + f(x_1, y_1, t_{k+1})]$  into a component of a vector function called  $\vec{f}$ .

When  $i = 2, j = 1$ , (Eq. 4.3) becomes

$$\begin{aligned}
& -\frac{\Delta t}{2\Delta x^2} \mathbf{u}_{3,1}^{k+1} + \left[ 1 + \frac{\Delta t}{\Delta x^2} + \frac{\Delta t}{\Delta y^2} \right] \mathbf{u}_{2,1}^{k+1} - \frac{\Delta t}{2\Delta x^2} \mathbf{u}_{1,1}^{k+1} - \frac{\Delta t}{2\Delta y^2} \mathbf{u}_{2,2}^{k+1} - \frac{\Delta t}{2\Delta y^2} \underbrace{\mathbf{u}_{2,0}^{k+1}}_{=\mathbf{u}_{2,1}^{k+1}} \\
& = \frac{\Delta t}{2\Delta x^2} \mathbf{u}_{3,1}^k + \left[ 1 - \frac{\Delta t}{\Delta x^2} - \frac{\Delta t}{\Delta y^2} \right] \mathbf{u}_{2,1}^k + \frac{\Delta t}{2\Delta x^2} \mathbf{u}_{1,1}^k + \frac{\Delta t}{2\Delta y^2} \mathbf{u}_{2,2}^k + \frac{\Delta t}{2\Delta y^2} \underbrace{\mathbf{u}_{2,0}^k}_{=\mathbf{u}_{2,1}^k} \\
& + \frac{\Delta t}{2} [f(x_2, y_1, t_k) + f(x_2, y_1, t_{k+1})]
\end{aligned}$$

To simplify, this is equivalent to



$$\begin{aligned}
& -\underbrace{\frac{\Delta t}{2\Delta x^2} u_{3,1}^{k+1}}_{A(2,3)} + \underbrace{\left[1 + \frac{\Delta t}{\Delta x^2} + \frac{\Delta t}{2\Delta y^2}\right]}_{A(2,2)} u_{2,1}^{k+1} - \underbrace{\frac{\Delta t}{2\Delta x^2} u_{1,1}^{k+1}}_{A(2,1)} - \underbrace{\frac{\Delta t}{2\Delta y^2} u_{2,2}^{k+1}}_{A(2,n+2)} \\
& = \underbrace{\frac{\Delta t}{2\Delta x^2} u_{3,1}^k}_{B(2,3)} + \underbrace{\left[1 - \frac{\Delta t}{\Delta x^2} - \frac{\Delta t}{2\Delta y^2}\right]}_{B(2,2)} u_{2,1}^k + \underbrace{\frac{\Delta t}{2\Delta x^2} u_{1,1}^k}_{B(2,1)} + \underbrace{\frac{\Delta t}{2\Delta y^2} u_{2,2}^k}_{B(2,n+2)} + \frac{\Delta t}{2} [f(x_2, y_1, t_k) + f(x_2, y_1, t_{k+1})]
\end{aligned}$$

When  $i = 3, j = 1$ , (Eq. 4.3) becomes

$$\begin{aligned}
& -\frac{\Delta t}{2\Delta x^2} u_{4,1}^{k+1} + \left[1 + \frac{\Delta t}{\Delta x^2} + \frac{\Delta t}{\Delta y^2}\right] u_{3,1}^{k+1} - \frac{\Delta t}{2\Delta x^2} u_{2,1}^{k+1} - \frac{\Delta t}{2\Delta y^2} u_{3,2}^{k+1} - \underbrace{\frac{\Delta t}{2\Delta y^2} u_{3,0}^{k+1}}_{=u_{3,1}^{k+1}} \\
& = \frac{\Delta t}{2\Delta x^2} u_{4,1}^k + \left[1 - \frac{\Delta t}{\Delta x^2} - \frac{\Delta t}{\Delta y^2}\right] u_{3,1}^k + \frac{\Delta t}{2\Delta x^2} u_{2,1}^k + \frac{\Delta t}{2\Delta y^2} u_{3,2}^k + \underbrace{\frac{\Delta t}{2\Delta y^2} u_{3,0}^k}_{=u_{3,1}^k} \\
& + \frac{\Delta t}{2} [f(x_3, y_1, t_k) + f(x_3, y_1, t_{k+1})]
\end{aligned}$$

To simplify, this is equivalent to

$$\begin{aligned}
& -\underbrace{\frac{\Delta t}{2\Delta x^2} u_{4,1}^{k+1}}_{A(3,4)} + \underbrace{\left[1 + \frac{\Delta t}{\Delta x^2} + \frac{\Delta t}{2\Delta y^2}\right]}_{A(3,3)} u_{3,1}^{k+1} - \underbrace{\frac{\Delta t}{2\Delta x^2} u_{2,1}^{k+1}}_{A(3,2)} - \underbrace{\frac{\Delta t}{2\Delta y^2} u_{3,2}^{k+1}}_{A(3,n+3)} \\
& = \underbrace{\frac{\Delta t}{2\Delta x^2} u_{4,1}^k}_{B(3,4)} + \underbrace{\left[1 - \frac{\Delta t}{\Delta x^2} - \frac{\Delta t}{2\Delta y^2}\right]}_{B(3,3)} u_{3,1}^k + \underbrace{\frac{\Delta t}{2\Delta x^2} u_{2,1}^k}_{B(3,2)} + \underbrace{\frac{\Delta t}{2\Delta y^2} u_{3,2}^k}_{B(3,n+3)} + \frac{\Delta t}{2} [f(x_3, y_1, t_k) + f(x_3, y_1, t_{k+1})]
\end{aligned}$$

When  $i = n, j = 1$ , (Eq. 4.3) becomes

$$\begin{aligned}
& -\frac{\Delta t}{2\Delta x^2} \underbrace{u_{n+1,1}^{k+1}}_{=u_{n,1}^{k+1}} + \left[1 + \frac{\Delta t}{\Delta x^2} + \frac{\Delta t}{\Delta y^2}\right] u_{n,1}^{k+1} - \frac{\Delta t}{2\Delta x^2} u_{n-1,1}^{k+1} - \frac{\Delta t}{2\Delta y^2} u_{n,2}^{k+1} - \underbrace{\frac{\Delta t}{2\Delta y^2} u_{n,0}^{k+1}}_{=u_{n,1}^{k+1}} \\
& = \frac{\Delta t}{2\Delta x^2} \underbrace{u_{n+1,1}^k}_{=u_{n,1}^k} + \left[1 - \frac{\Delta t}{\Delta x^2} - \frac{\Delta t}{\Delta y^2}\right] u_{n,1}^k + \frac{\Delta t}{2\Delta x^2} u_{n-1,1}^k + \frac{\Delta t}{2\Delta y^2} u_{n,2}^k + \underbrace{\frac{\Delta t}{2\Delta y^2} u_{n,0}^k}_{=u_{n,1}^k} \\
& + \frac{\Delta t}{2} [f(x_n, y_1, t_k) + f(x_n, y_1, t_{k+1})]
\end{aligned}$$

To simplify, this is equivalent to

$$\begin{aligned}
& \underbrace{\left[1 + \frac{\Delta t}{2\Delta x^2} + \frac{\Delta t}{2\Delta y^2}\right]}_{A(n,n)} \underbrace{u_{n,1}^{k+1}}_{A(n,n-1)} - \underbrace{\frac{\Delta t}{2\Delta x^2} u_{n-1,1}^{k+1}}_{A(n,n-1)} - \underbrace{\frac{\Delta t}{2\Delta y^2} u_{n,2}^{k+1}}_{A(n,n+n)} \\
&= \underbrace{\left[1 - \frac{\Delta t}{2\Delta x^2} - \frac{\Delta t}{2\Delta y^2}\right]}_{B(n,n)} \underbrace{u_{n,1}^k}_{B(n,n-1)} + \underbrace{\frac{\Delta t}{2\Delta x^2} u_{n-1,1}^k}_{B(n,n-1)} + \underbrace{\frac{\Delta t}{2\Delta y^2} u_{n,2}^k}_{B(n,n+n)} + \frac{\Delta t}{2} [f(x_n, y_1, t_k) + f(x_n, y_1, t_{k+1})]
\end{aligned}$$

Now  $i = 1, j = 2$ , (Eq. 4.3) becomes

$$\begin{aligned}
& -\frac{\Delta t}{2\Delta x^2} u_{2,2}^{k+1} + \left[1 + \frac{\Delta t}{\Delta x^2} + \frac{\Delta t}{\Delta y^2}\right] u_{1,2}^{k+1} - \frac{\Delta t}{2\Delta x^2} \underbrace{u_{0,2}^{k+1}}_{=u_{1,2}^{k+1}} - \frac{\Delta t}{2\Delta y^2} u_{1,3}^{k+1} - \frac{\Delta t}{2\Delta y^2} u_{1,1}^{k+1} \\
&= \frac{\Delta t}{2\Delta x^2} u_{2,2}^k + \left[1 - \frac{\Delta t}{\Delta x^2} - \frac{\Delta t}{\Delta y^2}\right] u_{1,2}^k + \frac{\Delta t}{2\Delta x^2} \underbrace{u_{0,2}^k}_{=u_{1,2}^k} + \frac{\Delta t}{2\Delta y^2} u_{1,3}^k + \frac{\Delta t}{2\Delta y^2} u_{1,1}^k \\
&+ \frac{\Delta t}{2} [f(x_1, y_2, t_k) + f(x_1, y_2, t_{k+1})]
\end{aligned}$$

To simplify, this is equivalent to

$$\begin{aligned}
& -\frac{\Delta t}{2\Delta x^2} u_{2,2}^{k+1} + \underbrace{\left[1 + \frac{\Delta t}{2\Delta x^2} + \frac{\Delta t}{\Delta y^2}\right]}_{A(n+1,n+1)} u_{1,2}^{k+1} - \underbrace{\frac{\Delta t}{2\Delta y^2} u_{1,3}^{k+1}}_{A(n+1,(n+1)+n)} - \underbrace{\frac{\Delta t}{2\Delta y^2} u_{1,1}^{k+1}}_{A(n+1,(n+1)-n)} \\
&= \frac{\Delta t}{2\Delta x^2} u_{2,2}^k + \underbrace{\left[1 - \frac{\Delta t}{2\Delta x^2} - \frac{\Delta t}{\Delta y^2}\right]}_{B(n+1,n+1)} u_{1,2}^k + \underbrace{\frac{\Delta t}{2\Delta y^2} u_{1,3}^k}_{B(n+1,(n+1)+n)} + \underbrace{\frac{\Delta t}{2\Delta y^2} u_{1,1}^k}_{B(n+1,(n+1)-n)} + \frac{\Delta t}{2} [f(x_1, y_2, t_k) + f(x_1, y_2, t_{k+1})]
\end{aligned}$$

The same arguments hold for  $i = 1, \dots, n$  and  $j = 3, \dots, m$ .

For  $i = 1, j = m$ , (Eq. 4.3) gives

$$\begin{aligned}
& -\frac{\Delta t}{2\Delta x^2} u_{2,m}^{k+1} + \left[1 + \frac{\Delta t}{\Delta x^2} + \frac{\Delta t}{\Delta y^2}\right] u_{1,m}^{k+1} - \frac{\Delta t}{2\Delta x^2} \underbrace{u_{0,m}^{k+1}}_{=u_{1,m}^{k+1}} - \frac{\Delta t}{2\Delta y^2} \underbrace{u_{1,m+1}^{k+1}}_{=u_{1,m}^{k+1}} - \frac{\Delta t}{2\Delta y^2} u_{1,m-1}^{k+1} \\
&= \frac{\Delta t}{2\Delta x^2} u_{2,m}^k + \left[1 - \frac{\Delta t}{\Delta x^2} - \frac{\Delta t}{\Delta y^2}\right] u_{1,m}^k + \frac{\Delta t}{2\Delta x^2} \underbrace{u_{0,m}^k}_{=u_{1,m}^k} + \frac{\Delta t}{2\Delta y^2} \underbrace{u_{1,m+1}^k}_{=u_{1,m}^k} + \frac{\Delta t}{2\Delta y^2} u_{1,m-1}^k \\
&+ \frac{\Delta t}{2} [f(x_1, y_m, t_k) + f(x_1, y_m, t_{k+1})]
\end{aligned}$$

To simplify, this becomes

$$\begin{aligned}
& \underbrace{-\frac{\Delta t}{2\Delta x^2}}_{A(n(m-1)+1, n(m-1)+2)} \mathbf{u}_{2,m}^{k+1} + \underbrace{\left[1 + \frac{\Delta t}{2\Delta x^2} + \frac{\Delta t}{2\Delta y^2}\right]}_{A(n(m-1)+1, n(m-1)+1)} \mathbf{u}_{1,m}^{k+1} - \underbrace{\frac{\Delta t}{2\Delta y^2}}_{A(n(m-1)+1, n(m-1)+1-n)} \mathbf{u}_{1,m-1}^{k+1} \\
= & \underbrace{\frac{\Delta t}{2\Delta x^2}}_{B(n(m-1)+1, n(m-1)+2)} \mathbf{u}_{2,m}^k + \underbrace{\left[1 - \frac{\Delta t}{2\Delta x^2} - \frac{\Delta t}{2\Delta y^2}\right]}_{B(n(m-1)+1, n(m-1)+1)} \mathbf{u}_{1,m}^k + \underbrace{\frac{\Delta t}{2\Delta y^2}}_{B(n(m-1)+1, n(m-1)+1-n)} \mathbf{u}_{1,m-1}^k \\
& + \frac{\Delta t}{2} [f(x_1, y_m, t_k) + f(x_1, y_m, t_{k+1})]
\end{aligned}$$

For  $i = 2, j = m$ , (Eq. 4.3) takes the form

$$\begin{aligned}
& -\frac{\Delta t}{2\Delta x^2} \mathbf{u}_{3,m}^{k+1} + \left[1 + \frac{\Delta t}{\Delta x^2} + \frac{\Delta t}{\Delta y^2}\right] \mathbf{u}_{2,m}^{k+1} - \frac{\Delta t}{2\Delta x^2} \mathbf{u}_{1,m}^{k+1} - \underbrace{\frac{\Delta t}{2\Delta y^2} \mathbf{u}_{2,m+1}^{k+1}}_{=\mathbf{u}_{2,m}^{k+1}} - \frac{\Delta t}{2\Delta y^2} \mathbf{u}_{2,m-1}^{k+1} \\
= & \frac{\Delta t}{2\Delta x^2} \mathbf{u}_{3,m}^k + \left[1 - \frac{\Delta t}{\Delta x^2} - \frac{\Delta t}{\Delta y^2}\right] \mathbf{u}_{2,m}^k + \frac{\Delta t}{2\Delta x^2} \mathbf{u}_{1,m}^k + \frac{\Delta t}{2\Delta y^2} \underbrace{\mathbf{u}_{2,m+1}^k}_{=\mathbf{u}_{2,m}^k} + \frac{\Delta t}{2\Delta y^2} \mathbf{u}_{2,m-1}^k \\
& + \frac{\Delta t}{2} [f(x_2, y_m, t_k) + f(x_2, y_m, t_{k+1})]
\end{aligned}$$

To simplify, this becomes

$$\begin{aligned}
& \underbrace{-\frac{\Delta t}{2\Delta x^2}}_{A(n(m-1)+2, n(m-1)+3)} \mathbf{u}_{3,m}^{k+1} + \underbrace{\left[1 + \frac{\Delta t}{\Delta x^2} + \frac{\Delta t}{2\Delta y^2}\right]}_{A(n(m-1)+2, n(m-1)+2)} \mathbf{u}_{2,m}^{k+1} - \underbrace{\frac{\Delta t}{2\Delta x^2}}_{A(n(m-1)+2, n(m-1)+1)} \mathbf{u}_{1,m}^{k+1} - \underbrace{\frac{\Delta t}{2\Delta y^2}}_{A(n(m-1)+2, n(m-1)+2-n)} \mathbf{u}_{2,m-1}^{k+1} \\
= & \underbrace{\frac{\Delta t}{2\Delta x^2}}_{B(n(m-1)+2, n(m-1)+3)} \mathbf{u}_{3,m}^k + \underbrace{\left[1 - \frac{\Delta t}{\Delta x^2} - \frac{\Delta t}{2\Delta y^2}\right]}_{B(n(m-1)+2, n(m-1)+2)} \mathbf{u}_{2,m}^k + \underbrace{\frac{\Delta t}{2\Delta x^2}}_{B(n(m-1)+2, n(m-1)+1)} \mathbf{u}_{1,m}^k + \underbrace{\frac{\Delta t}{2\Delta y^2}}_{B(n(m-1)+2, n(m-1)+2-n)} \mathbf{u}_{2,m-1}^k \\
& + \frac{\Delta t}{2} [f(x_2, y_m, t_k) + f(x_2, y_m, t_{k+1})]
\end{aligned}$$

Finally, when  $i = n, j = m$ , (Eq. 4.3) becomes

$$\begin{aligned}
& -\frac{\Delta t}{2\Delta x^2} \underbrace{u_{n+1,m}^{k+1}}_{=u_{n,m}^{k+1}} + \left[ 1 + \frac{\Delta t}{\Delta x^2} + \frac{\Delta t}{\Delta y^2} \right] u_{n,m}^{k+1} - \frac{\Delta t}{2\Delta x^2} u_{n-1,m}^{k+1} - \frac{\Delta t}{2\Delta y^2} \underbrace{u_{n,m+1}^{k+1}}_{=u_{n,m}^{k+1}} - \frac{\Delta t}{2\Delta y^2} u_{n,m-1}^{k+1} \\
& = \frac{\Delta t}{2\Delta x^2} \underbrace{u_{n+1,m}^k}_{=u_{n,m}^k} + \left[ 1 - \frac{\Delta t}{\Delta x^2} - \frac{\Delta t}{\Delta y^2} \right] u_{n,m}^k + \frac{\Delta t}{2\Delta x^2} u_{n-1,m}^k + \frac{\Delta t}{2\Delta y^2} \underbrace{u_{n,m+1}^k}_{=u_{n,m}^k} + \frac{\Delta t}{2\Delta y^2} u_{n,m-1}^k \\
& + \frac{\Delta t}{2} [f(x_n, y_m, t_k) + f(x_n, y_m, t_{k+1})]
\end{aligned}$$

To simplify, this becomes

$$\begin{aligned}
& \underbrace{\left[ 1 + \frac{\Delta t}{2\Delta x^2} + \frac{\Delta t}{2\Delta y^2} \right]}_{A(n-m,n-m)} u_{n,m}^{k+1} - \underbrace{\frac{\Delta t}{2\Delta x^2}}_{A(n-m,n-m-1)} u_{n-1,m}^{k+1} - \underbrace{\frac{\Delta t}{2\Delta y^2}}_{A(n-m,n-m-n)} u_{n,m-1}^{k+1} \\
& = \underbrace{\left[ 1 - \frac{\Delta t}{2\Delta x^2} - \frac{\Delta t}{2\Delta y^2} \right]}_{B(n-m,n-m)} u_{n,m}^k + \underbrace{\frac{\Delta t}{2\Delta x^2}}_{B(n-m,n-m-1)} u_{n-1,m}^k + \underbrace{\frac{\Delta t}{2\Delta y^2}}_{B(n-m,n-m-n)} u_{n,m-1}^k + \frac{\Delta t}{2} [f(x_n, y_m, t_k) + f(x_n, y_m, t_{k+1})]
\end{aligned}$$

We can put all the components of  $A$  and  $B$  together to obtain the two matrices  $A_{(n-m,n-m)}$  and  $B_{(n-m,n-m)}$ .

After obtaining  $A_{nm \times nm} \vec{u}_{nm \times 1}^{k+1} = B_{nm \times nm} \vec{u}_{nm \times 1}^k + \vec{f}_{nm \times 1} = \vec{b}_{nm \times 1}$ , we solve it in MATLAB by  $\vec{u}^{k+1} = A \setminus \vec{b}$ .

We use (Eq. 4.1) and the parameters in Table 4 to show the temperature rise of a model problem with the numerical points  $n = m = 128$  in Figure 20. The MATLAB code is attached in Appendix D.

Input	Value	Unit
$\alpha_T$	1	$\text{m}^2/\text{s}$
$K_T$	1	$\text{W}/(\text{m}^*\text{k})$
$I_0$	1	$\text{W}/\text{m}^3$
$d$	0.02	m
$L_x$	1	m
$L_y$	1	m
$x_0$	0.5	m
$a$	0.25	m
$b$	0.25	m
<i>period</i>	1	s

Table 4. 2-D film MATLAB input parameters for rotating laser beam

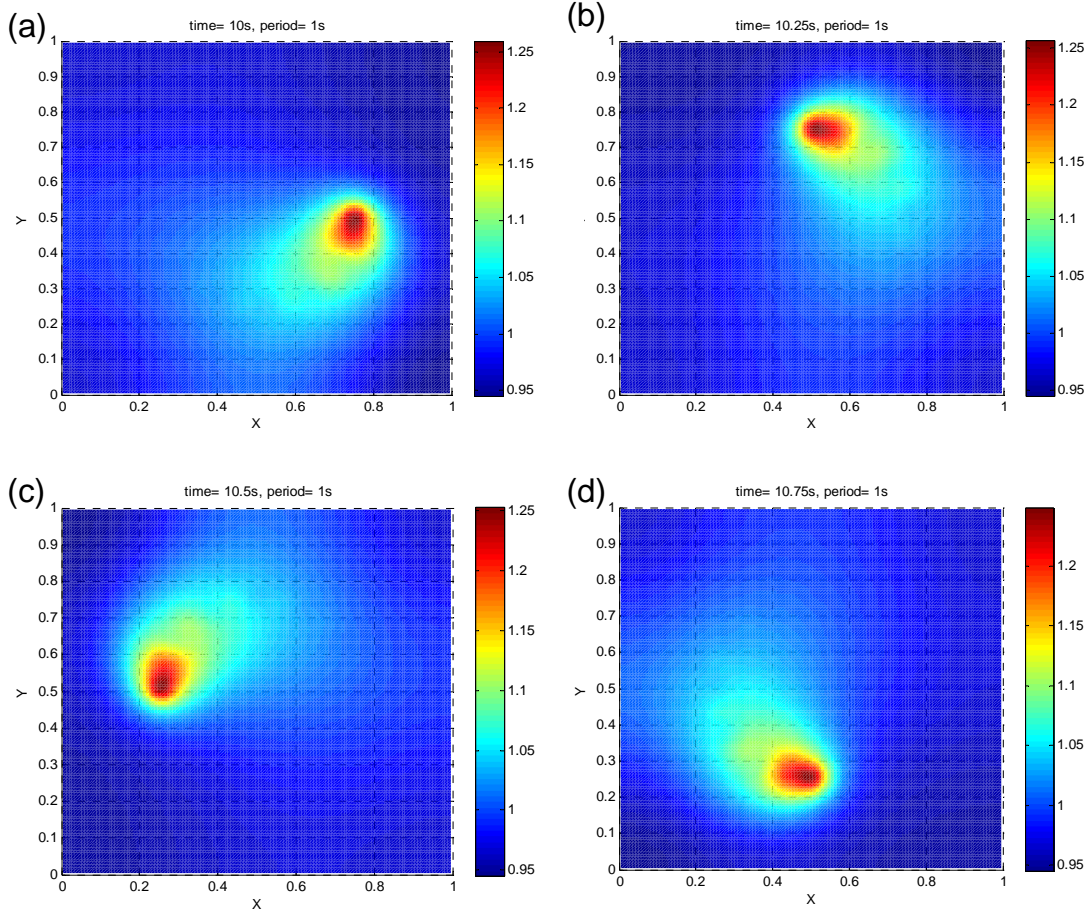
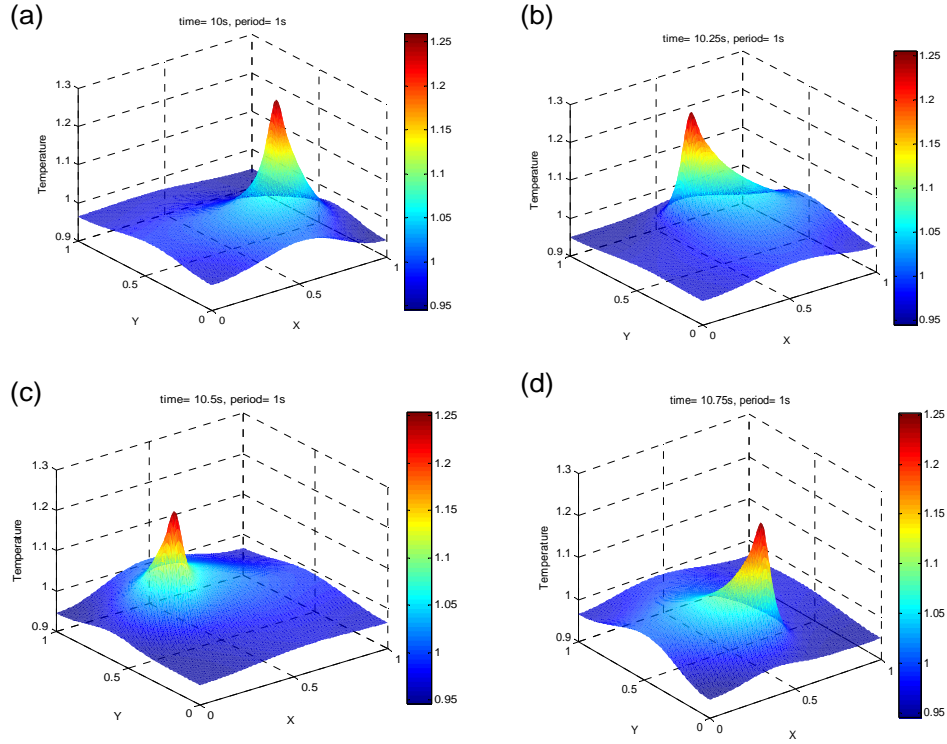


Figure 20. Snapshots of the temperature rise on a film of a model problem induced by a rotating Gaussian beam using the 2-D Crank-Nicolson method

## B. THE FAST FOURIER TRANSFORM (FFT) METHOD

The structures of Fast Fourier Transform in 1-D and 2-D are similar; instead of using `fft` and inverse Fast Fourier Transform `ifft` commands in 1-D MATLAB code, the 2-D code uses `fft2` and `ifft2` to carry out the computation. Briefly speaking, one needs to even extend the problem from domain  $[0,1] \times [0,1]$  to  $[0,2] \times [0,2]$ , make the problem periodic in both  $x$  and  $y$  direction with period 2, and then apply the Fourier transform and its inverse to obtain the numerical solution. Our MATLAB code is attached in Appendix E.

Figure 21 uses the input parameters from Table 4 and returns a very similar result to Figure 20. The 2D FFT has a better performance than the 2-D Crank-Nicolson method. In other words, FFT can produce the result faster than the Crank-Nicolson method with the same number of numerical points. We will discuss the details in Section D.



(Figure continued on next page.)

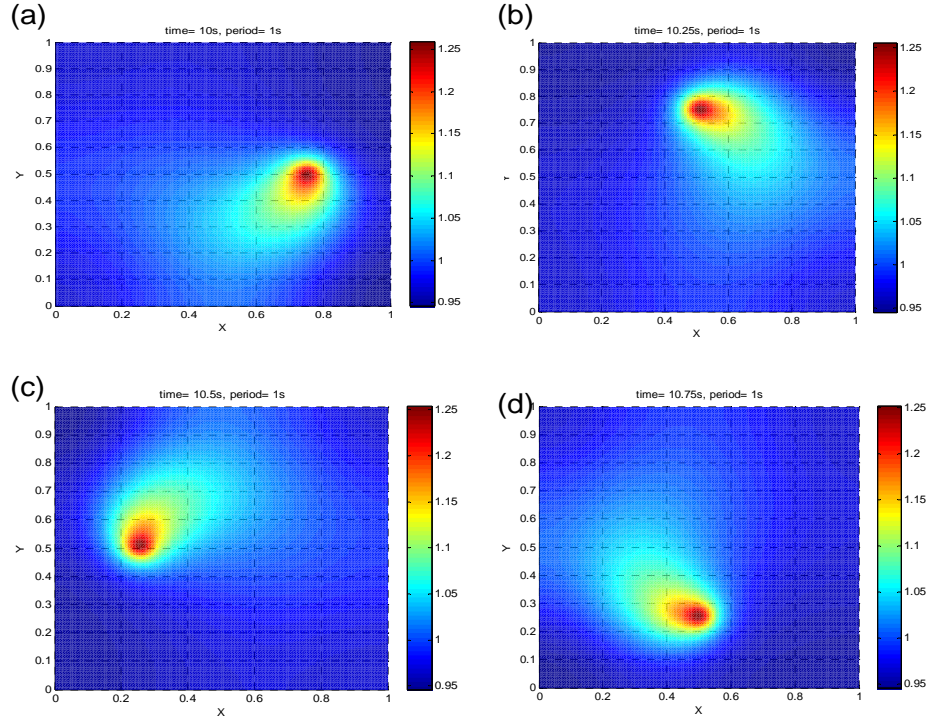


Figure 21. Snapshots of the temperature rise on a film of a model problem induced by a rotating Gaussian beam using 2-D FFT method in 3-D view

### C. COMSOL

Figure 22 (a) is the result from COMSOL and Figure 22 (b) is the result using the Crank-Nicolson method in MATALB based on (Eq. 4.1) and Table 4 with the numerical points  $n = m = 128$  in  $x$  and  $y$  direction, respectively. The results are very close and we will do a point-by-point error analysis in Section D. The detailed process of creating this 2-D COMOSL is attached in Appendix F.



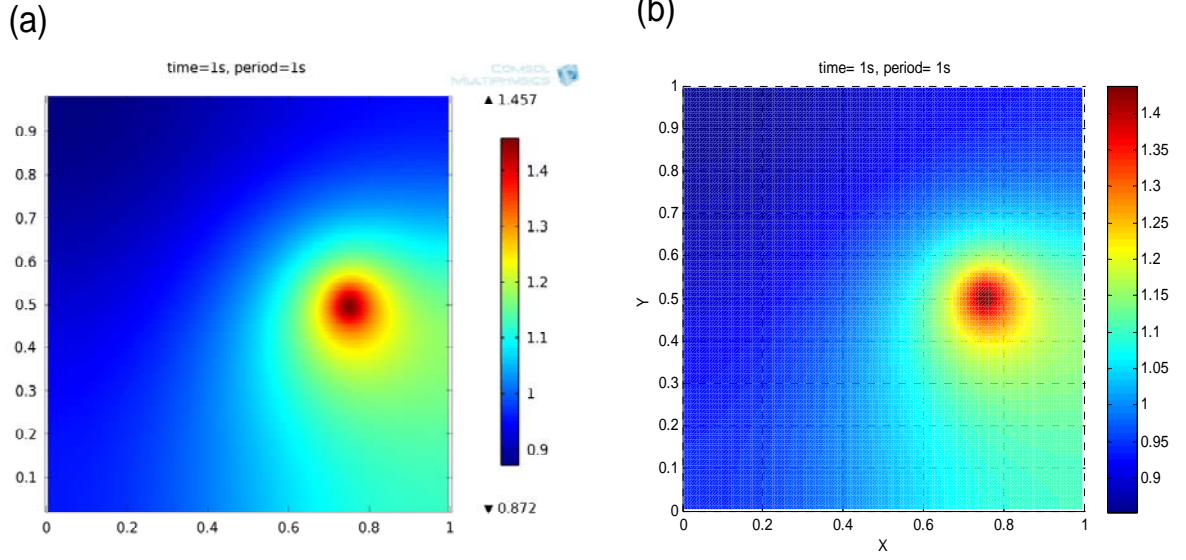


Figure 22. Model problem comparison: (a) COMSOL (b) MATLAB FFT 2 method

#### D. COMPARISON ON A MODEL PROBLEM

As we have mentioned in Section B, FFT returns the solution faster than Crank-Nicolson; the efficiency comparison is shown in Figure 23. When  $N=2^9$  points, we can see that FFT takes about  $10^2$  seconds but Crank-Nicolson takes more than  $10^3$  seconds to finish the computation. So FFT is about ten times faster than the Crank-Nicolson method for this test problem.

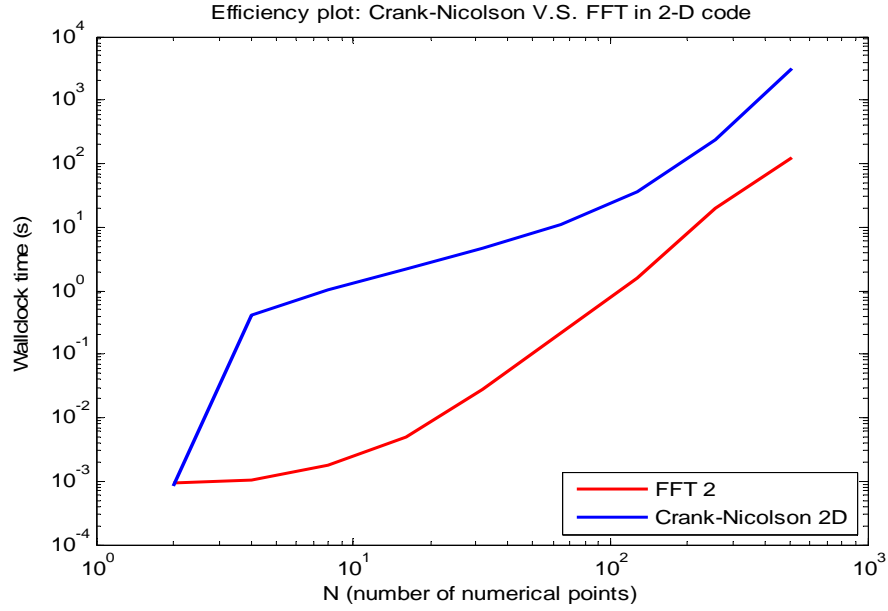


Figure 23. Efficiency plot: Crank-Nicolson 2-D method versus FFT 2-D method

In Figure 24, by using the result from Figure 22, we compare the relative difference in temperature rise with respect to time at a fixed point ( $x=0.5$ ,  $y=0.5$ ) in MATALB FFT 2 method and COMSOL using number of point  $N=256$ . The relative error is about  $10^{-3}$  and this result is tolerable. The relative error goes down as we increase the number of numerical grid points  $N$ .

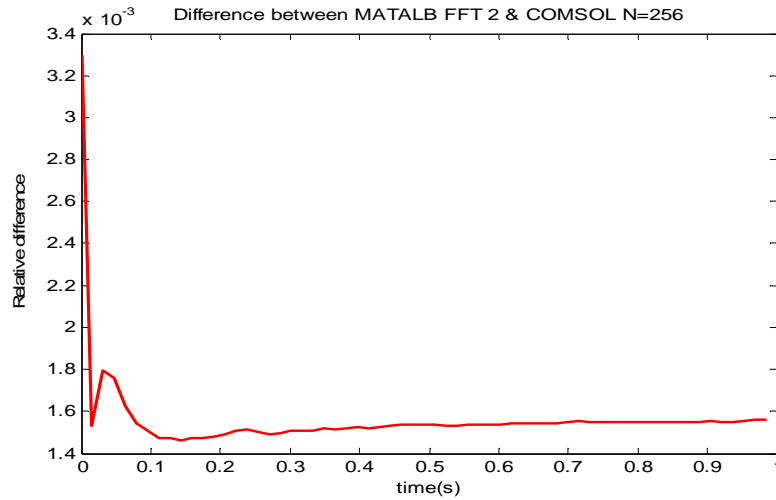


Figure 24. Relative error plot in 2-D FFT method and COMSOL

### E. REAL PROBLEM SIMULATION (STEEL AISI 4340)

We use Steel AISI 4340 as our test material with material properties from Table 2 and the rotating laser we deploy has the following input in Table 5:

Input	Value	Unit
$I_0$	1.0e8	W/m <sup>3</sup>
$d$	0.02	m
$L_x$	1	m
$L_y$	1	m
$x_0$	0.5	m
$y_0$	0.5	m
$a$	0.25	m
$b$	0.25	m
$period$	1	s

Table 5. 2-D rotating laser input on Steel AISI 4340.

Figure 25 depicts the temperature rise at different times within one period. It is observed that heat will not spread out quickly enough due to the properties of the Steel AISI 4340 material so the temperature at those points directly shined by the laser rise quickly. However, those points far away from the points hit by laser, for instance, the center point ( $x=0.5, y=0.5$ ), has almost no temperature rise.

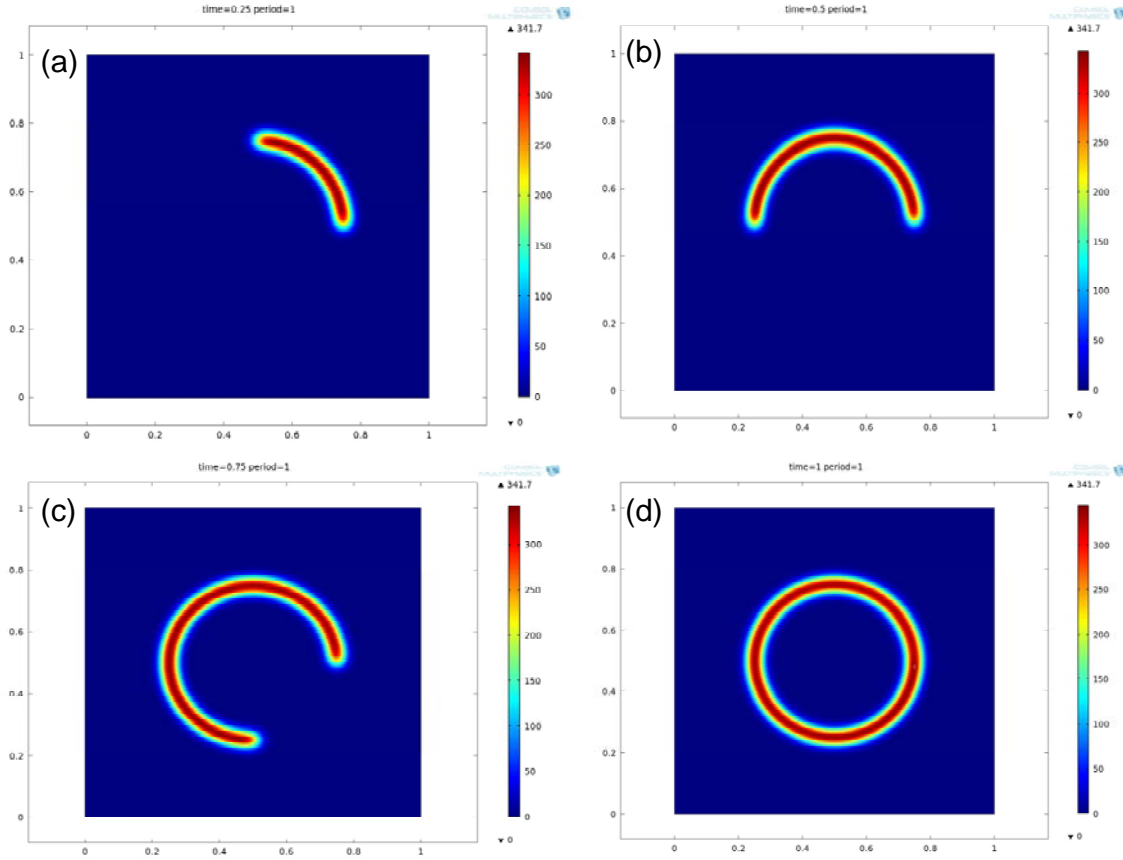


Figure 25. Snapshots of the temperature rise on a Steel AISI 4340 film induced by a rotating Gaussian beam using COMOSL with period=1s

Figure 26 shows the temperature rise as a function of time at a fixed point ( $x=0.75$ ,  $y=0.5$ ) of Steel AISI 4340 with dithering laser beam period=0.1 sec. The temperature increases as the laser beam rotates close to the point and the temperature almost stays steady as the laser beam moves away. The overall line shape behaves as an increasing function of time, as expected.

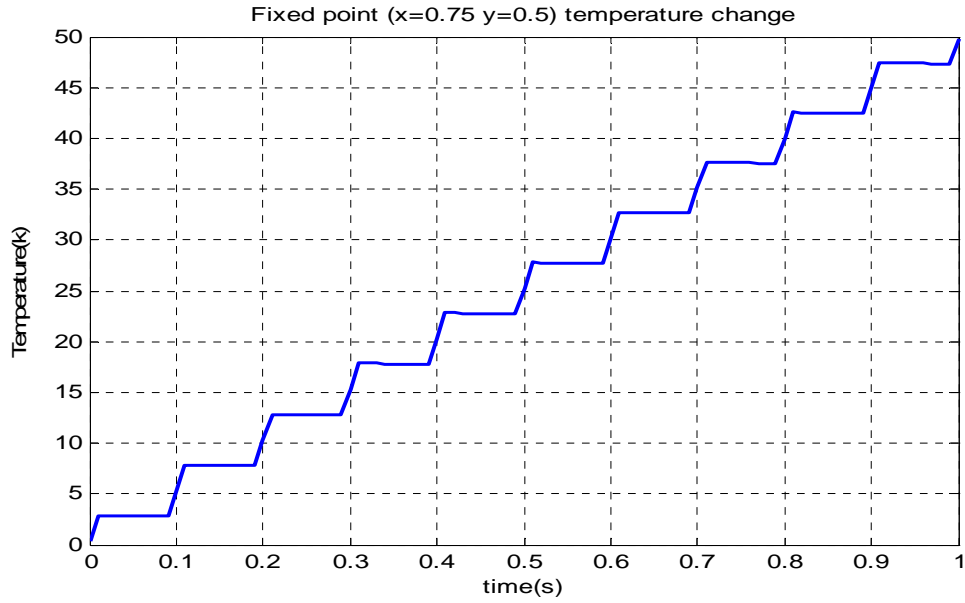


Figure 26. 2-D temperature change with time at a fixed point of Steel AISI 4340 with rotating beam period=0.1s

The quantitative relationship between the maximum temperature rise and the rotating frequency at time=1s is depicted in Figure 27; the maximum temperature rise is a decreasing function of the frequency (reciprocal of the rotating period).

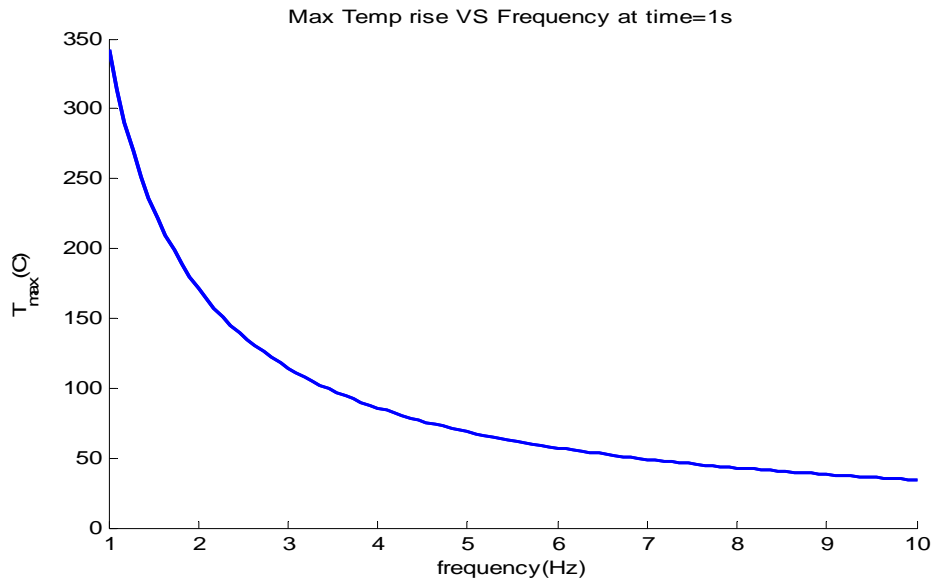


Figure 27. 2-D maximum temperature rise of Steel AISI 4340 versus the frequency of the rotating laser beam

THIS PAGE INTENTIONALLY LEFT BLANK

## V. NUMERICAL SOLUTION FOR A TRANSIENT, THREE-DIMENSIONAL TEMPERATURE DISTRIBUTION IN A FINITE SOLID DUE TO A ROTATING OR DITHERING LASER BEAM

We have already verified that COMSOL has returned a very accurate solution compared with other numerical methods in both 1-D and 2-D codes. Therefore, instead of writing a complicated MATLAB 3-D code, we use COMSOL to obtain the 3-D answer. Recall (Eq. 4.1) and impose insulated boundary conditions. We use Steel AISI 4340 as our test material with material properties from Table 2 and the rotating laser we deploy has the following input from Table 6:

Input	Value	Unit
$I_0$	5.0e5	W/m <sup>2</sup>
$d$	0.02	m
$L_x$	1	m
$L_y$	1	m
$L_z$	1	m
$x_0$	0.5	m
$y_0$	0.5	m
$a$	0.25	m
$b$	0.25	m
$period$	1	s

Table 6. 3-D rotating laser input on Steel AISI 4340

Figure 28 depicts the temperature rise at different times within one period. The hottest spot is where the point hit instantly by the laser as in the 1-D and 2-D cases. The

maximum temperature rise is 1468K. Those points far away from the area hit by the laser, for instance, the center point ( $x=0.5, y=0.5, z=1$ ), have little temperature rise.

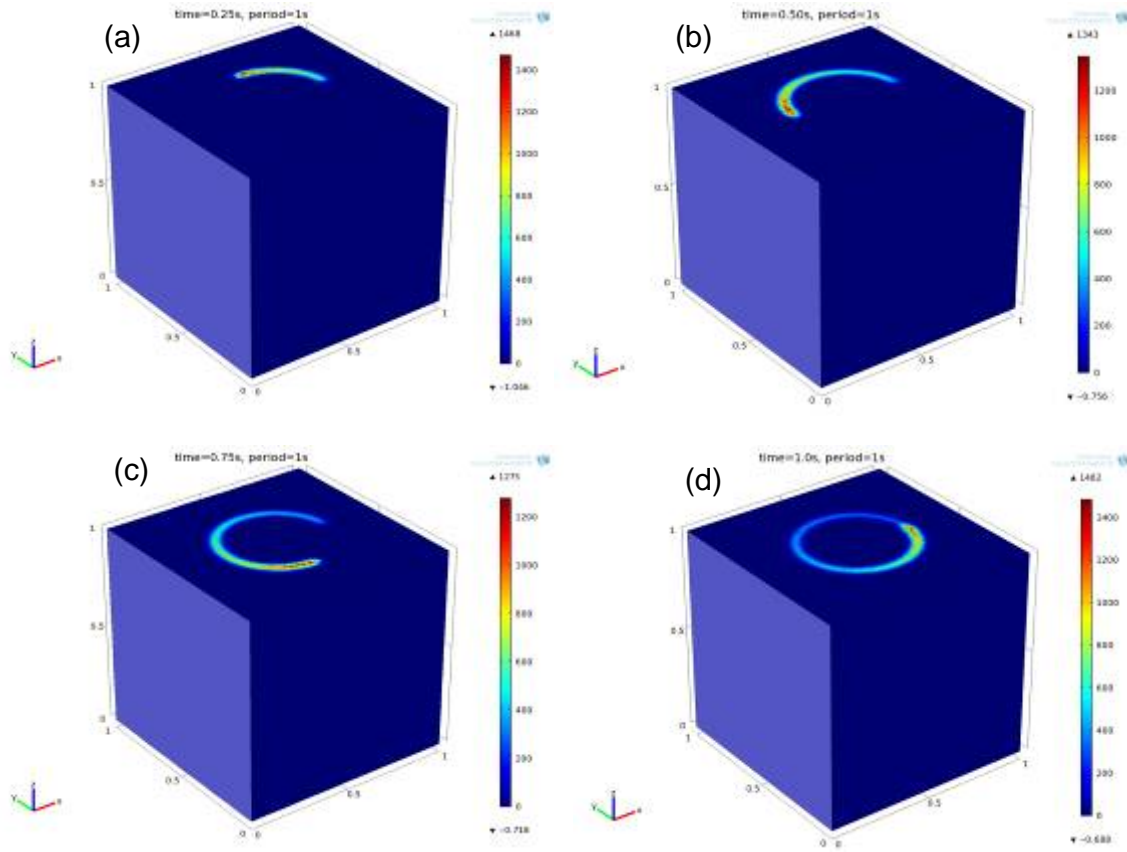


Figure 28. Snapshots of the temperature rise on a Steel AISI 4340 solid induced by a rotating Gaussian beam using COMSOL with period=1s

Figure 29 shows the temperature rise at a fixed point ( $x=0.75, y=0.5, z=1$ ) as a function of time. Figure 30 shows the maximum temperature rise of the whole domain as a function of time; the overall hottest spot is at ( $x=0.341, y=0.307, z=1$ ) when time=0.64s with the rotating laser beam period=1s. Figure 31 and Figure 32 depict the temperature rise at fixed time=1s in different layers. It is observed that the heat does not spread downward quickly and there is almost no temperature rise 0.1m below the top surface shined by the laser. After several tryouts, materials have larger values in diffusivity than Steel AISI 4340 can make heat spread out faster and so make the temperature rise higher than Steel AISI 4340.



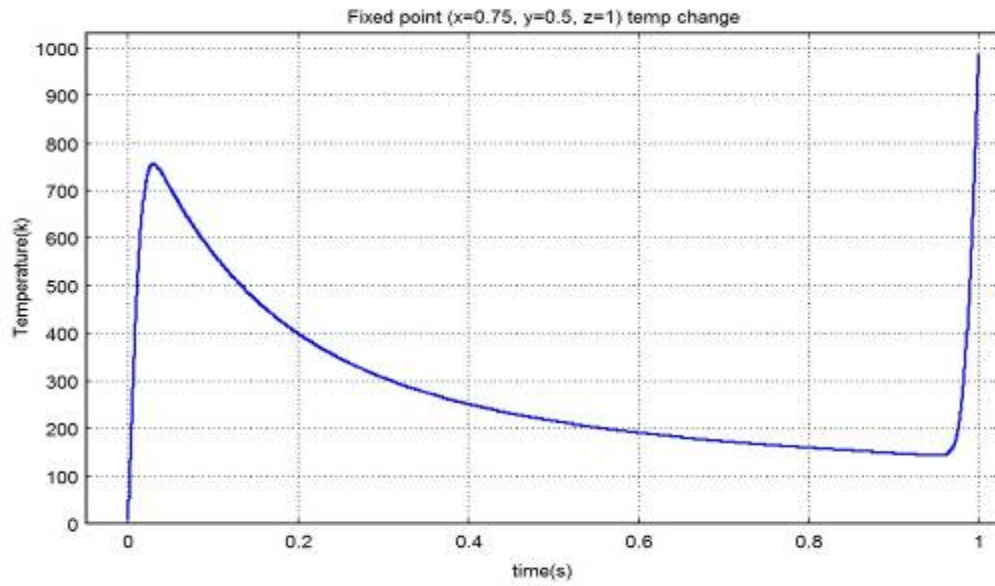


Figure 29. 3-D temperature change as a function of time at a fixed point of steel AISI 4340 with rotating laser beam period=1s

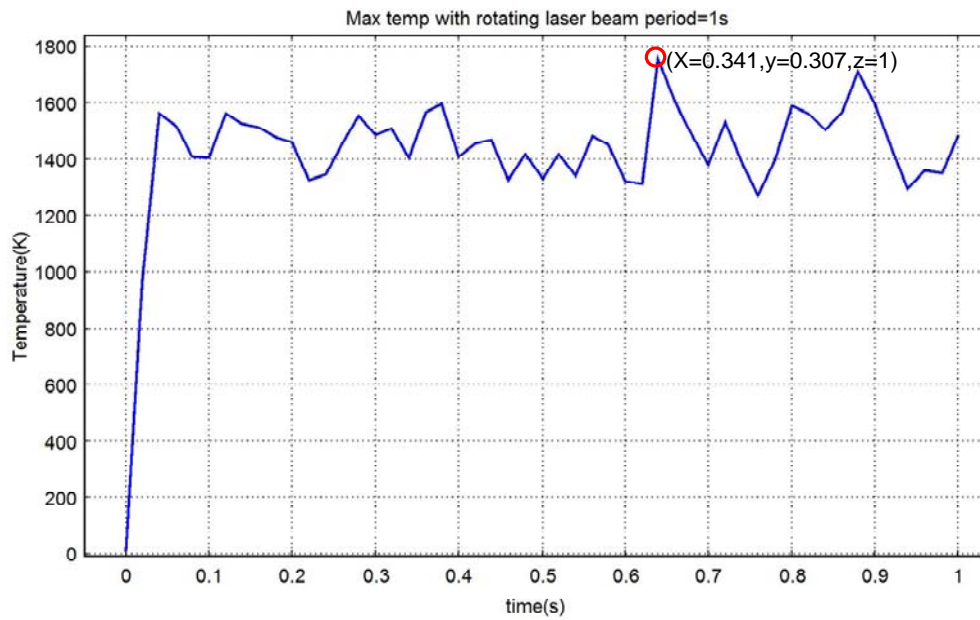


Figure 30. Maximum temperature as a function of time with rotating laser beam period =1s

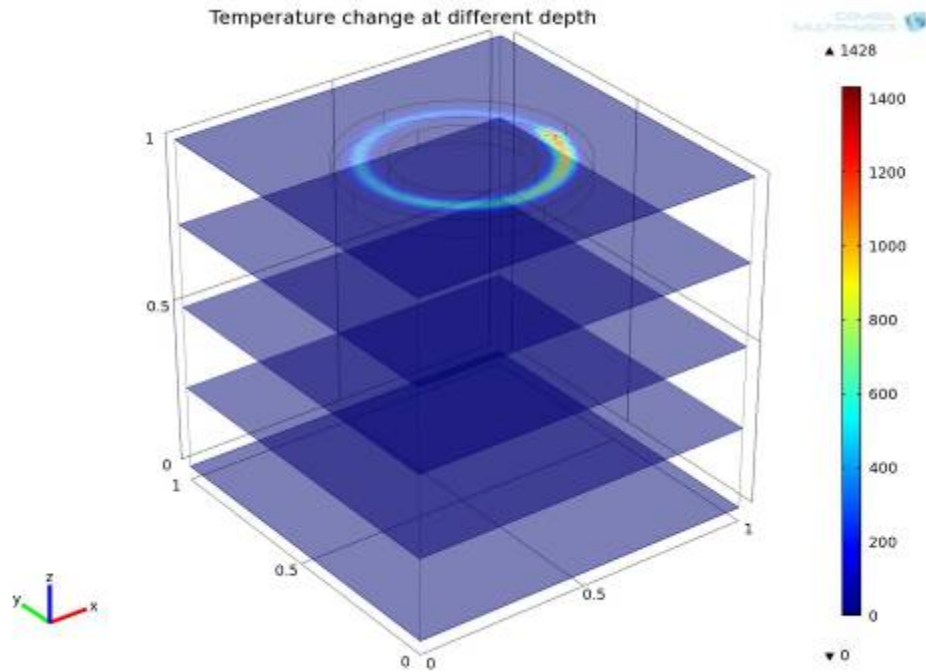


Figure 31. 3-D temperature rise at different layers from  $z=0$  to  $z=1$

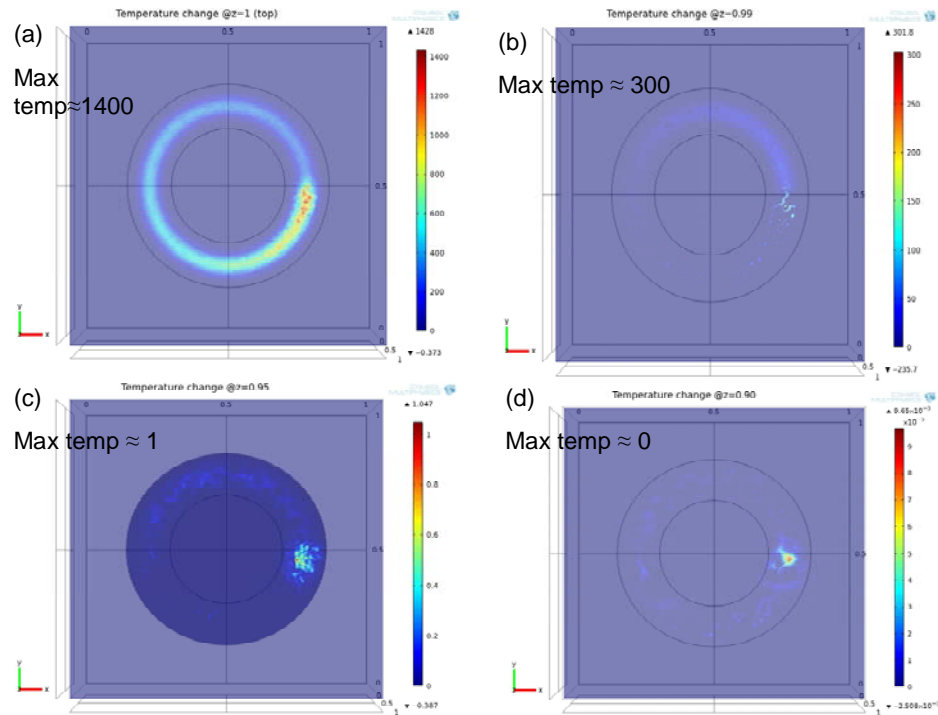


Figure 32. Maximum temperature at different depth (a) $z=1$  top surface (b) $z=0.99$  (c) $z=0.95$  (d) $z=0.90$

Figure 33 shows the maximum temperature rise at time=1s when the period is reduced to 0.1s; the laser beam rotates 10 cycles. The maximum temperature rise decreases from 1754K of period=1s to 670K of period=0.1s. Therefore, the maximum temperature rise can be reduced by increasing the frequency of the rotating laser beam.

Figure 34 shows the temperature rise at a fixed point ( $x=0.75, y=0.5, z=1$ ) of period=0.1s as a function of time. The overall temperature rise oscillates, and its envelope behaves as an increasing function of time, but its maximum temperature rise is smaller compared to the maximum temperature rise with period=1s, as illustrated in Figure 29. Figure 35 shows the maximum temperature rise of the whole domain as a function of time, and its envelope behaves as an increasing function as well. Figure 36 depicts the quantitative relationship between the maximum temperature rise and the rotating frequency at time=1s. The results agree with earlier analytical studies for the semi-infinite domain [5].

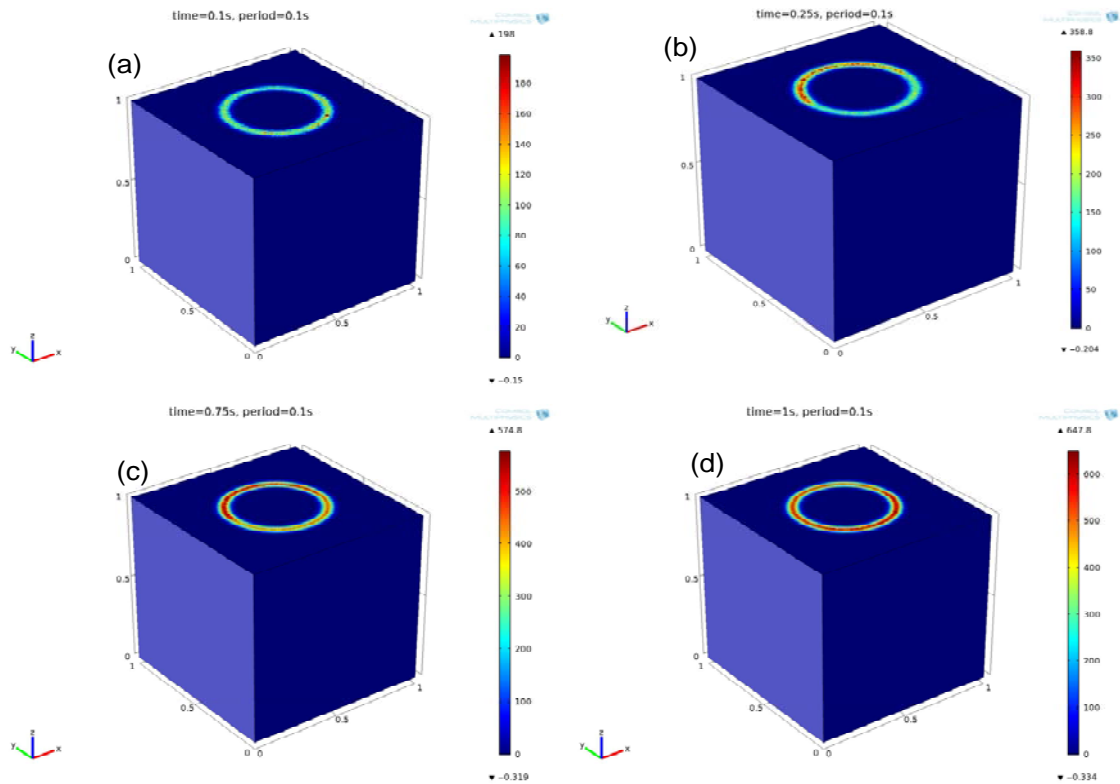


Figure 33. Snapshots of the temperature rise on a Steel AISI 4340 solid induced by a rotating Gaussian beam using COMSOL with period=0.1s

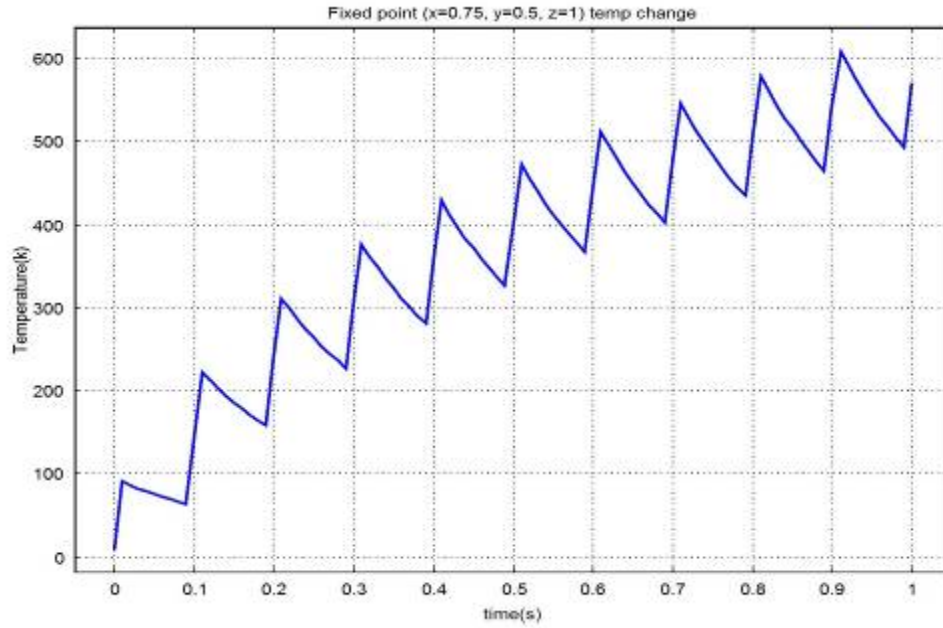


Figure 34. 3-D temperature change as a function of time at a fixed point of Steel AISI 4340 with rotating laser beam period=0.1

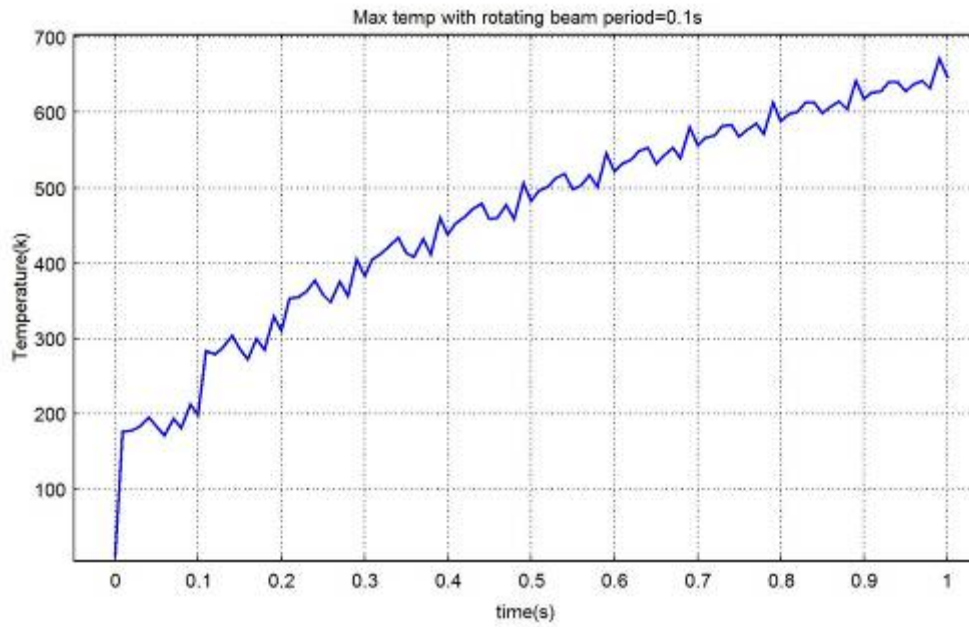


Figure 35. Maximum temperature as a function of time with rotating laser beam period =0.1s

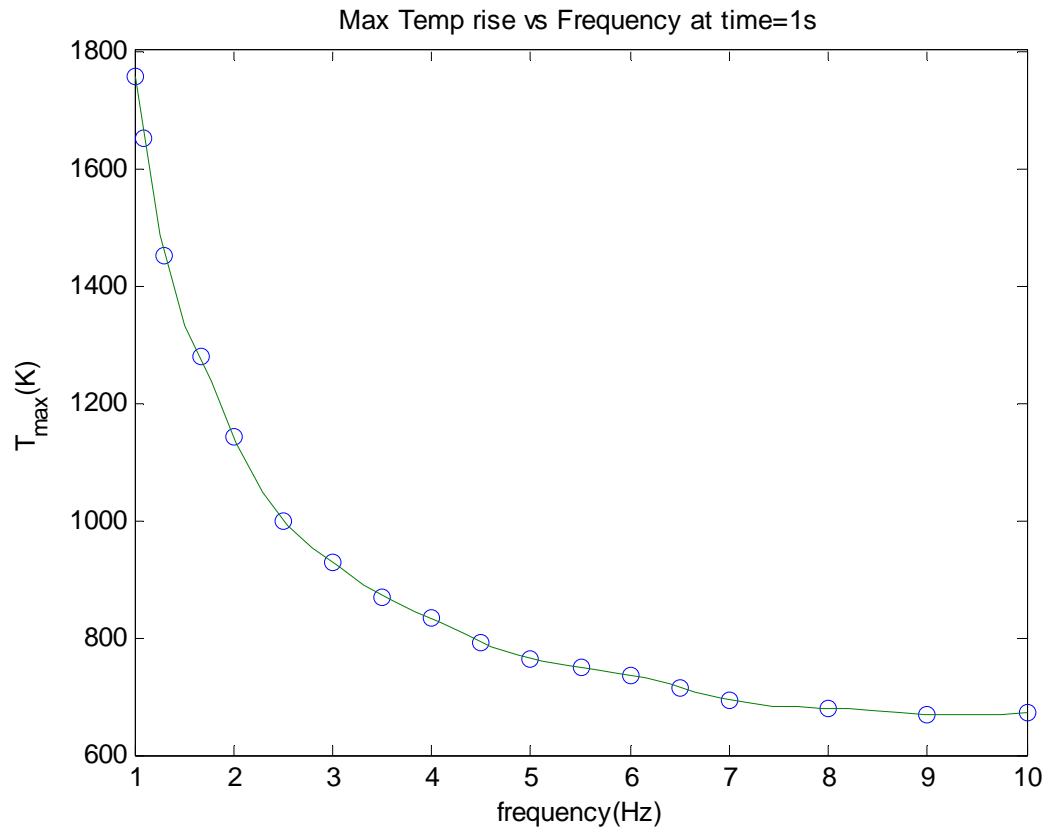


Figure 36. 3-D maximum temperature rise of Steel AISI 4340 versus the frequency of the rotating laser beam

THIS PAGE INTENTIONALLY LEFT BLANK

## VI. CONCLUSIONS AND FUTURE WORK

In this thesis, both analytical and numerical solutions for describing the transient temperature rise induced by a moving laser in a finite domain have been developed. We have exploited several methods including the eigenfunction expansion, the Crank-Nicolson scheme, FFT and COMSOL.

We have confirmed that the faster the laser rotates (i.e., the higher the frequency) the lower the temperature rise induced. In other words, to reduce the military's vulnerability to high-energy laser weapons it is possible to let the object rotate or rock to minimize the temperature rise. The quantitative relationship between maximum temperature rise and laser beam rotation frequency can be used as a general guide for adjusting the speed of rotation of the object in order to prevent temperature rise from reaching the melting point.

This thesis can be explored deeper in the future. Some future potentials endeavors include but not limited to:

- A. Increase or decrease the effective radius of the laser beam  $d$  in Figure 3 and Figure 17 to analyze how temperature rise is affected.
- B. Increase or decrease the radii  $a$  and  $b$  of the rotating trajectory of the Gaussian beam in Tables 3, 5 and 6 to analyze how maximum temperature rise is affected.
- C. Create 3-D MATLAB codes and compare the results with COMSOL.
- D. Instead of using  $TEM_{00}$  mode Gaussian distribution as the heat source illustrated in Figure 17, different transversal modes in a laser spot such as  $TEM_{01}$  or  $TEM_{11}$ , can be used to further seek the analytical and numerical solutions [8].
- E. Thermal properties are assumed to be temperature dependent: this makes the nonhomogeneous heat equation nonlinear, but more realistic [3]. This

nonlinear model can be solved without mathematical background in nonlinear programming using appendix G, COMSOL code for 3-D simulation by imposing certain materials whose thermal properties are temperature-dependent, such as silver [9].

- F. In COMSOL 3-D geometry, try cylindrical coordinate and spherical coordinate rather than Cartesian coordinate.
- G. Conduct an experiment and see if the theoretical modeling is accurate.



## APPENDIX A. CRANK-NICOLSON CODE FOR 1-D SIMULATION

```

N=256; % number of numerical points in [0, 1]
dx=1/N; % spatial step
x=(1:N)'/N; % numerical grid
u=zeros(N,1); % solution at current time, u(j)=u(x(j))
d=0.02; % radius of Gaussian source
a=1.0; % magnitude of Gaussian source
u0=zeros(N,1); % initial value
u=u0;
%
dt=dx/8; % time step
T=25/256;
m=T/dt; % number of time steps
t=[0:m]*dt;
%
r=dt/dx^2;
% forming the matrices
d0=[0.5; ones(N-2,1); 0.5]*r;
d1=0.5*ones(N,1)*r;
d_l=0.5*ones(N,1)*r;
A=spdiags([-d_l, 1+d0, -d1],[-1,0,1],N,N);
B=spdiags([d_l, 1-d0, d1],[-1,0,1],N,N);
%
y0=0.5+0.25*sin(10*2*pi*t(1)); % location of Gaussian source at t^{k-1}
f0=a*exp(-(x-y0).^2/(2*d^2))/sqrt(2*pi*d^2); % Gaussian source at t^{k-1}
%
plot(x,u,'b-','linewidth',2.0)
axis([0,1,-0.04,0.16])
drawnow
for k=1:m,
    y1=0.5+0.25*sin(10*2*pi*t(k+1)); % location of Gaussian source at t^k
    f1=a*exp(-(x-y1).^2/(2*d^2))/sqrt(2*pi*d^2); % Gaussian source at t^k
    b=B*u+dt*(f0+f1)/2; % right-hand-side of the linear eq
    for u^{k}
        u=A\b; % solving the linear eq for u^{k}
        f0=f1;
        %pause(0.1)
        plot(x,u,'b-','linewidth',2.0)
        axis([0,1,-0.04,0.16])
        drawnow
    end
end

```

THIS PAGE INTENTIONALLY LEFT BLANK

## APPENDIX B. FFT CODE FOR 1-D SIMULATION


```

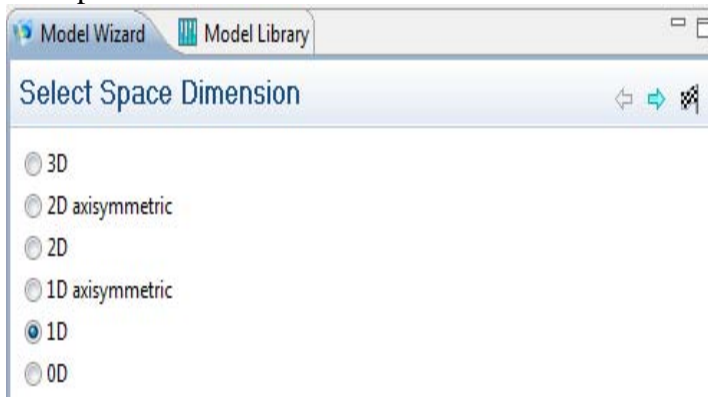
N=256;                % number of numerical points in [0, 1]
dx=1/N;               % spatial step
x=[0:N] * dx;         % numerical grid
u=zeros(N+1,1);       % solution at current time, u(j)=u(x(j))
d=0.02;               % radius of Gaussian source
a=1;                  % magnitude of Gaussian source
u0=zeros(N+1,1);      % initial value
u=u0;
%
dt=dx/8;              % time step
T=25/256;
m=T/dt;               % number of time steps
t=[0:m]*dt;
%
% going to the coefficients of cosin expansion
w=[u; u(N:-1:2)];
z=fft(w);
cu=real(z(1:N+1))/N;
r=[0:N] * pi.^2;
h1=[dt; (1-exp(-r(2:N+1)*dt))./r(2:N+1)];
h2=[0.5*dt.^2; (1-(r(2:N+1)*dt+1).*exp(-r(2:N+1)*dt))./r(2:N+1).^2];
%
y0=0.5+0.25*sin(10*2*pi*0); % location of Gaussian source at t^{k-1}
f0=a*exp(-(x-y0).^2/(2*d.^2))/sqrt(2*pi*d.^2); % Gaussian source at t^{k-1}
w=[f0; f0(N:-1:2)];
z=fft(w);
cf0=real(z(1:N+1))/N;
%
plot(x,u, 'r-', 'linewidth', 2.0)
axis([0,1,-0.04,0.16])
drawnow
for k=1:m,
    y1=0.5+0.25*sin(10*2*pi*t(k+1)); % location of Gaussian source at t^k
    f1=a*exp(-(x-y1).^2/(2*d.^2))/sqrt(2*pi*d.^2); % Gaussian source at t^k
    w=[f1; f1(N:-1:2)];
    z=fft(w);
    cf1=real(z(1:N+1))/N;
% update cu
    cu=cu.*exp(-r*dt)+cf1.*h1+(cf0-cf1)/dt.*h2;
    cf0=cf1;
% going back to the function
    z=N*[cu; cu(N:-1:2)];
    w=ifft(z);
    u=real(w(1:N+1));
    %pause(0.1)
    plot(x,u, 'r-', 'linewidth', 2.0)
    axis([0,1,-0.04,0.16])
    drawnow
end


```

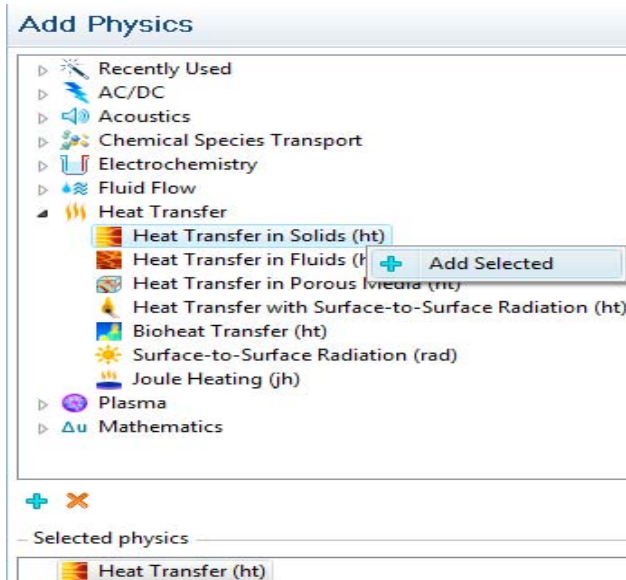
THIS PAGE INTENTIONALLY LEFT BLANK


## APPENDIX C. COMSOL CODE FOR 1-D SIMULATION

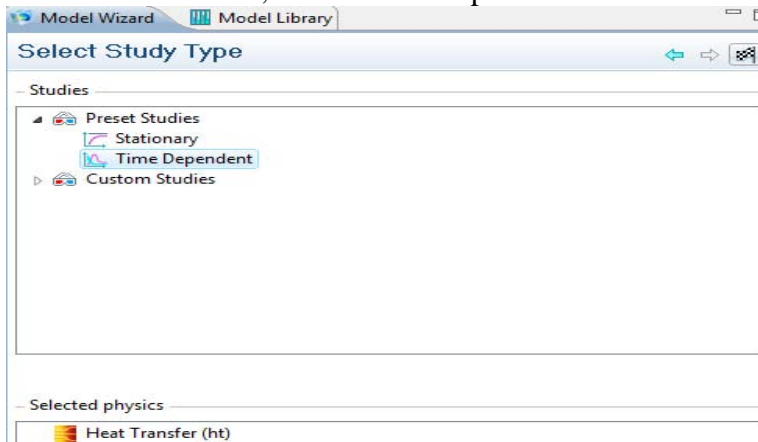
1. Open COMSOL 4.0a with 1D and hit .



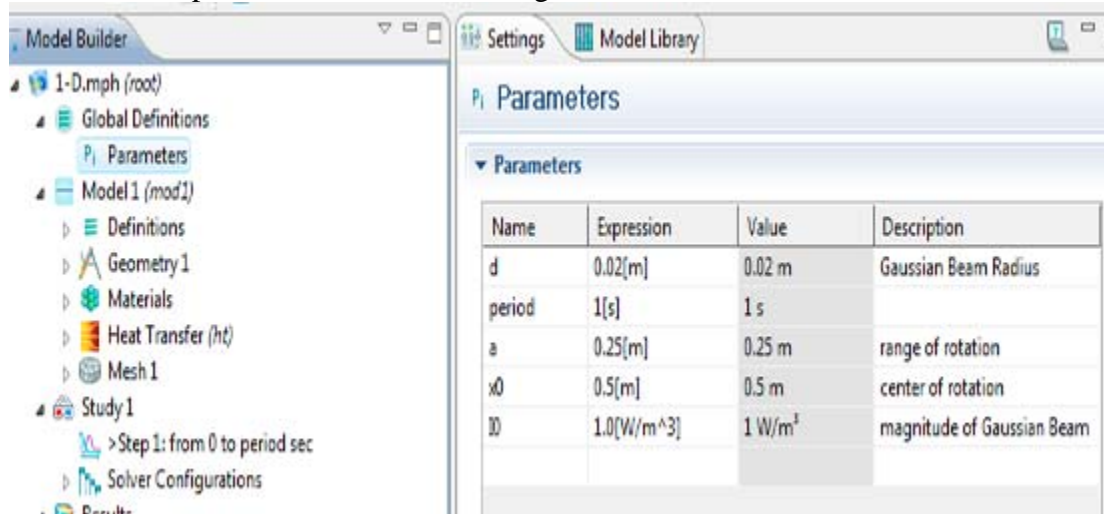
2. In Heat Transfer, select Heat Transfer in Solids (ht) and right-click Add selected then hit .



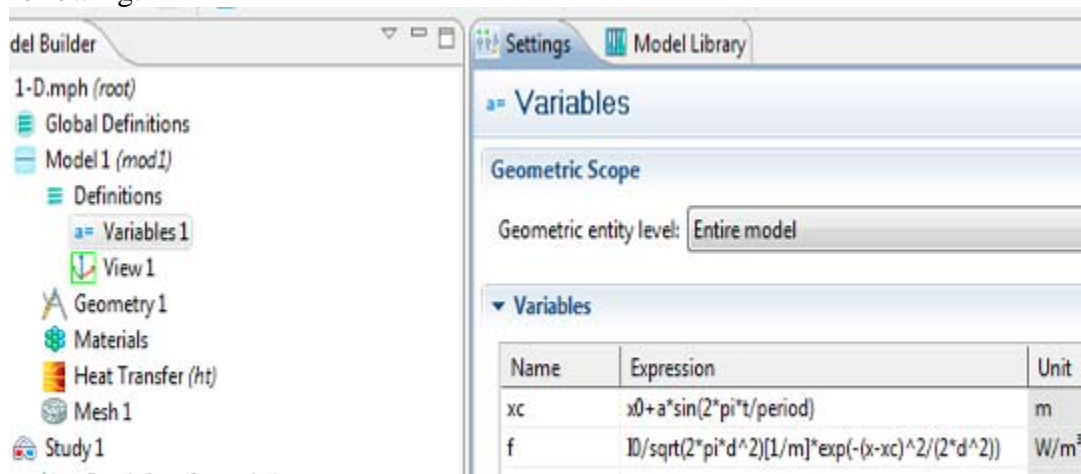
3. In Preset Studies, select Time Dependent and hit Finish .




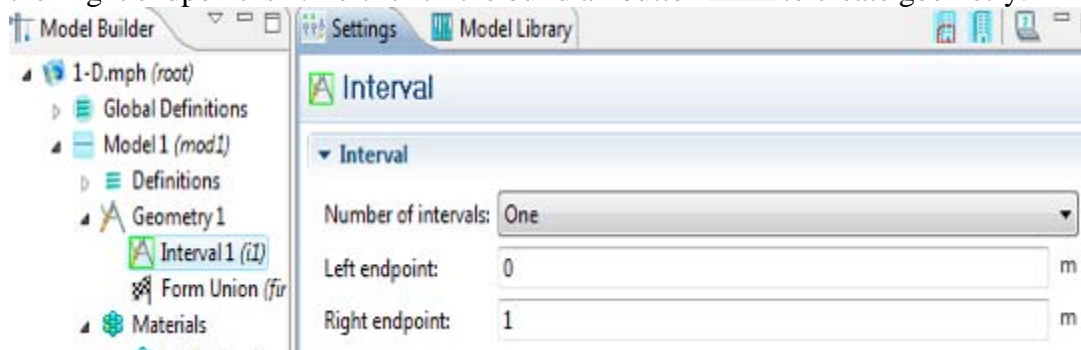
4. Under Model Builder, right-click on Global Definitions and left-click to select Parameters, input Parameters as following:



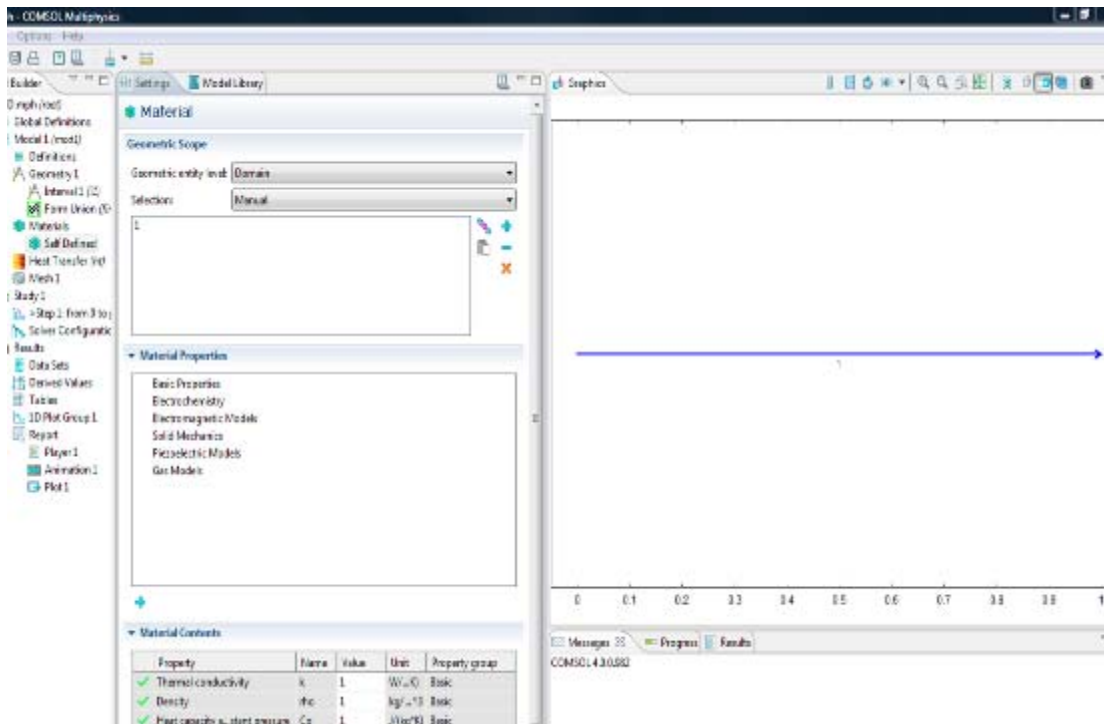
5. Right-click on Definitions and left click to select Variables, input Variables as following:



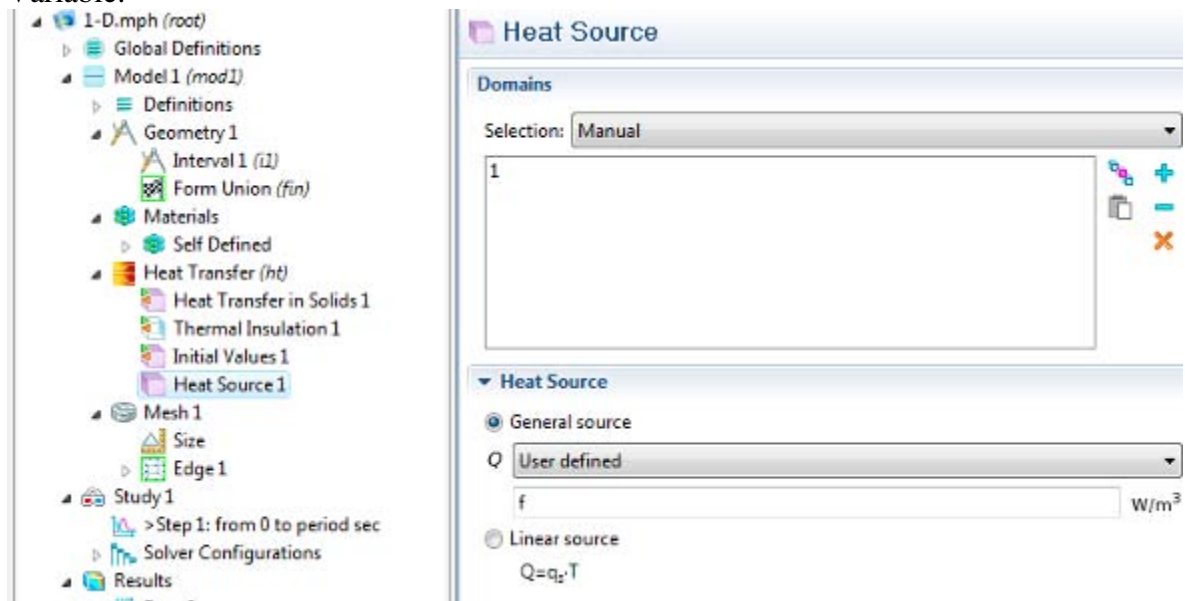
6. Right-click on Geometry and left-click to add Interval. The Left endpoint is 0 and the Right endpoint is 1. Left- click the build all button  to create geometry.



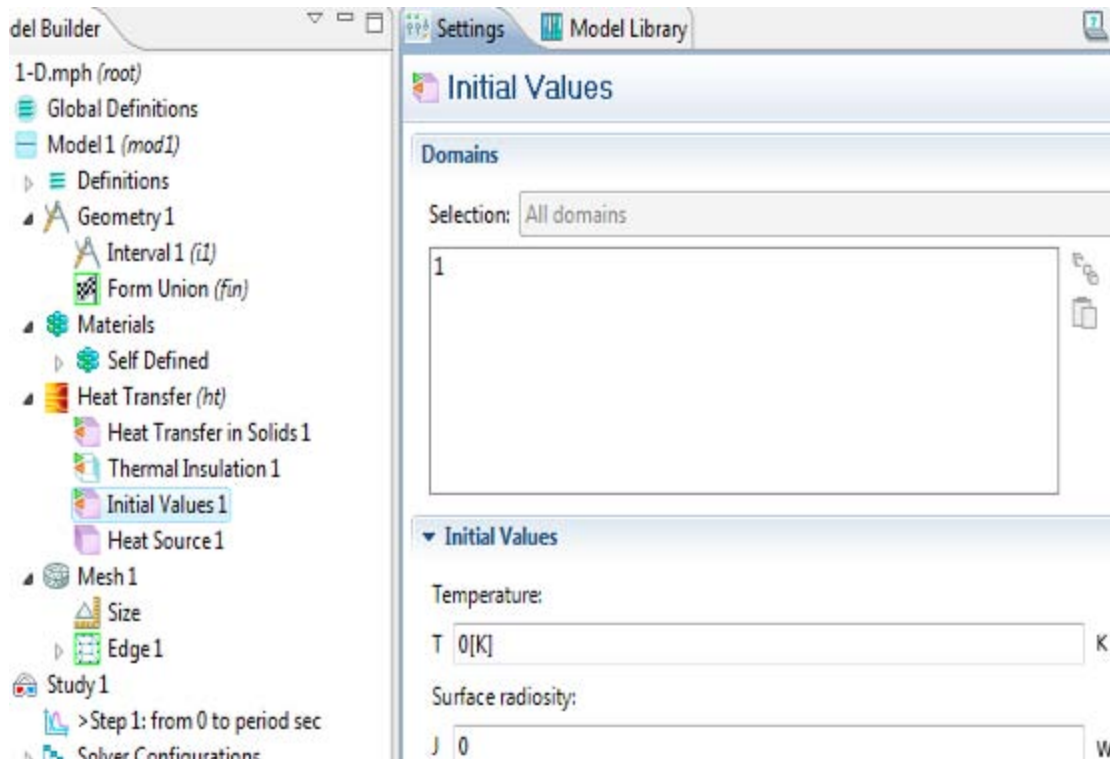
7. Right-click on Materials and left-click on add Materials. Make all the values in Material Contents equal 1. Right-click on the “line” in Graphics and left-click on it to select. Make sure that “1” is under the selection.




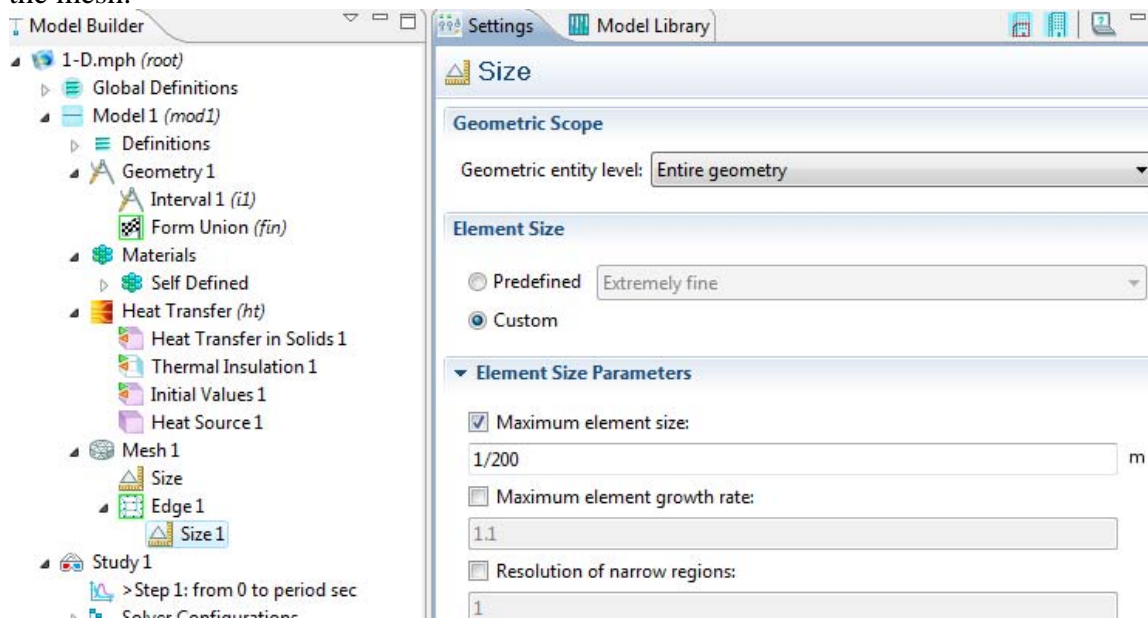
8. Right-click on Heat Transfer and left-click to add Heat Source. For General source Q, Select User defined and Put “f”, which is the heat source defined from the Variable.




9. In Initials Values, make Temperature equal 0.

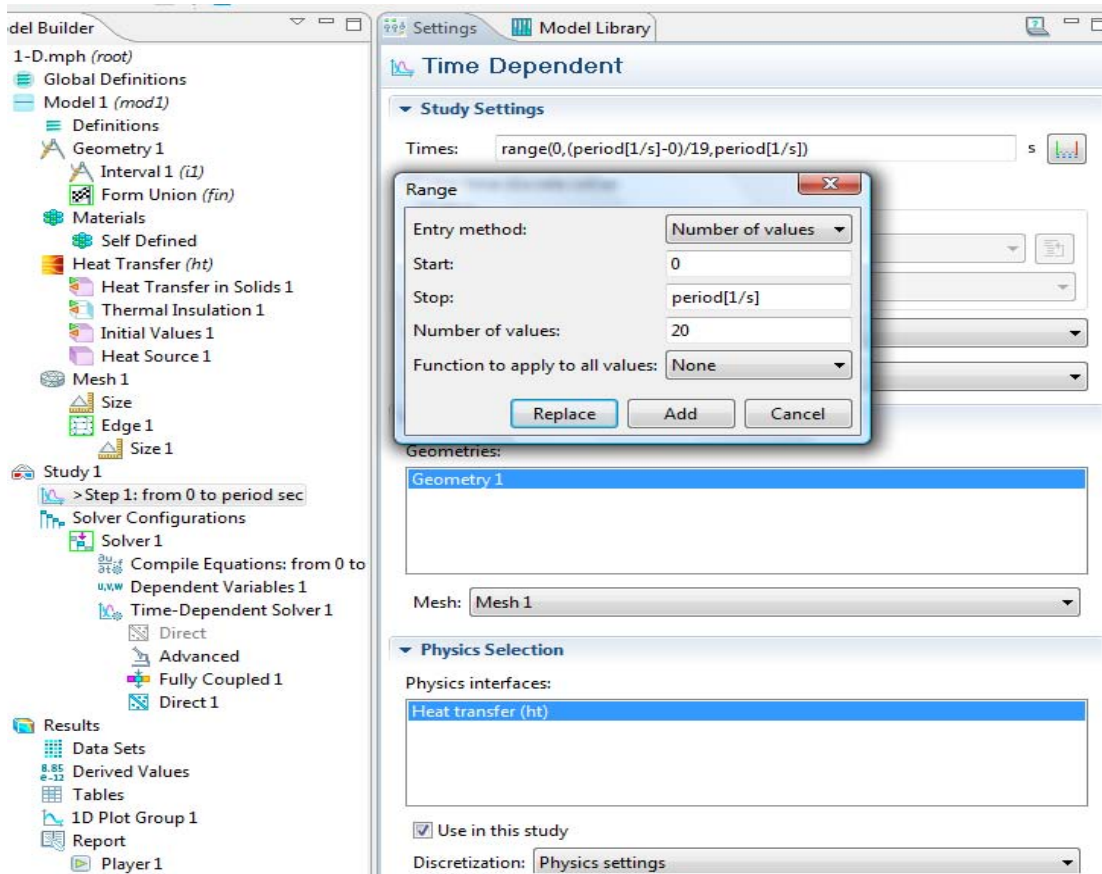


10. Right-click on Mesh and select Edge, in Edge under Element Size, select Custom and make Maximum element size equal  $1/200$ . Then select the build all button  to build the mesh.

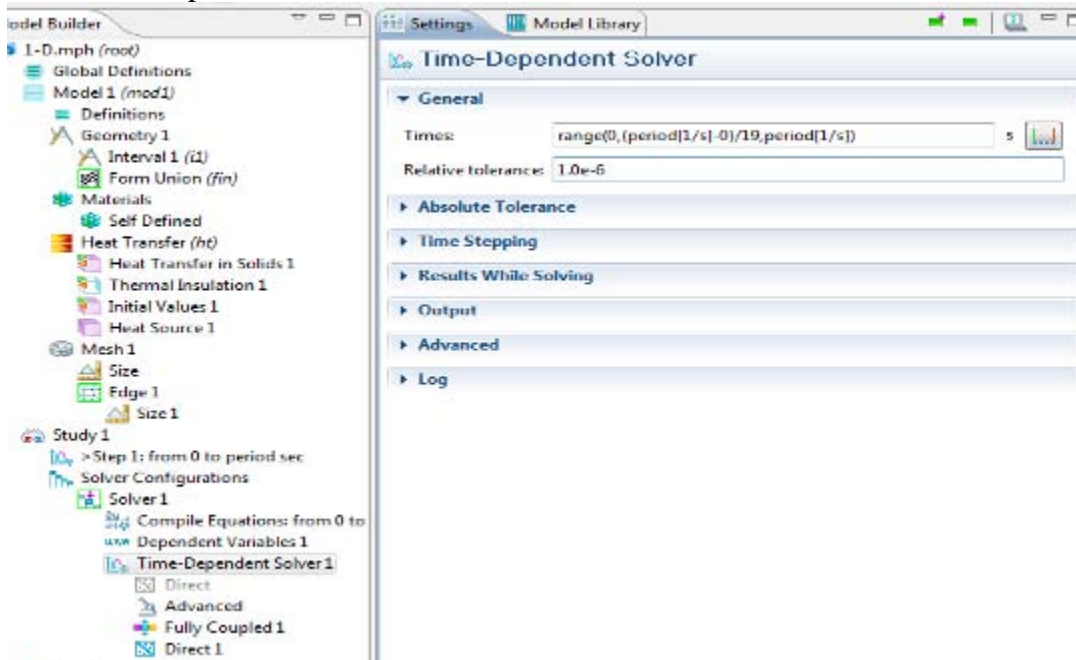


11. Under Study in Step, select Range button , under Entry method, select Number of values, start from 0 and stop at “period” which is defined in the parameters. Put Number of values to be 20.



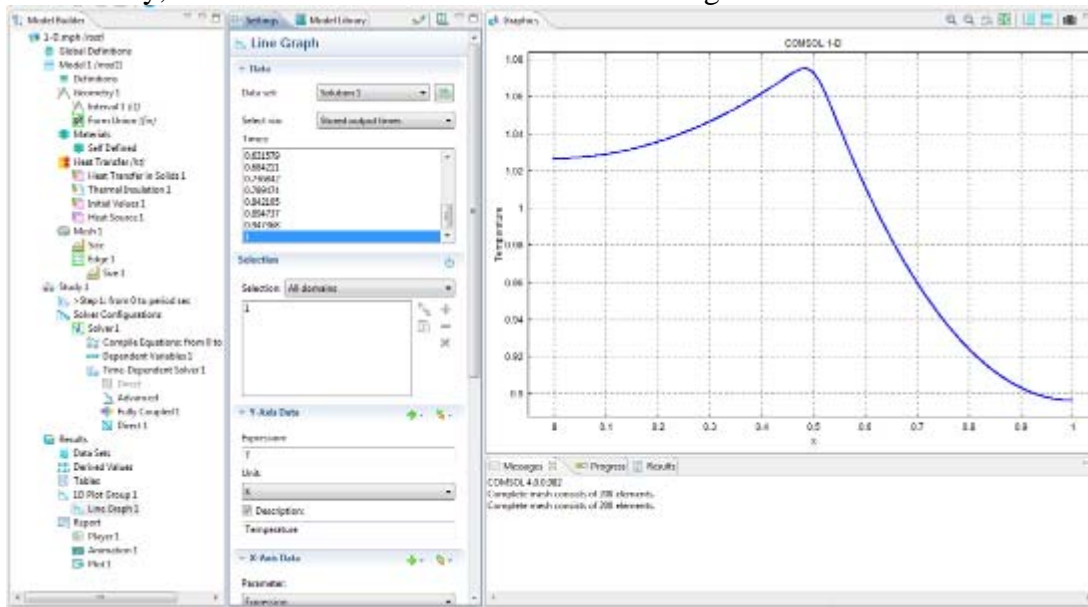


12. Under Time-Dependent Solver, make Relative tolerance to be 1.0e-6. Left Click on  to compute.



13. Finally, under Results, select Line Graph under 1D Plot Group and select time to be

1 only, we should be able to see the result like Figure 9.



14. Further data analysis can be done under Report, such as generate a movie and export data and further compare results with MATLAB.

## APPENDIX D. CRANK-NICOLSON CODE FOR 2-D SIMULATION

```
N=128; % number of numerical points in x direction
M=128; % number of numerical points in y direction
dx=1/N; % spatial step
dy=1/M;
x=(1:N)'/-0.5)*dx; % numerical grid
y=(1:M)'/-0.5)*dy;
[xx,yy]=meshgrid(x,y);
u=zeros(M*N,1); % solution at current time
d=0.02; % radius of Gaussian source
a=1; % magnitude of Gaussian source
u0=zeros(M*N,1); % initial value
u=u0;
%
dt=0.5*dx; % time step
m=11; % number of time steps
t=[0:m]*dt;
%
r1=dt/dx^2;
r2=dt/dy^2;
%
A = sparse([],[],[],N*M,N*M);
B = sparse([],[],[],N*M,N*M);
%% forming the matrices
% the bottom boundary
k=1;
A(k,k)=1+r1/2+r2/2;
A(k,k+1)=-r1/2;
A(k,k+N)=-r2/2;
B(k,k)=1-r1/2-r2/2;
B(k,k+1)=r1/2;
B(k,k+N)=r2/2;
%
k=N;
A(k,k)=1+r1/2+r2/2;
A(k,k-1)=-r1/2;
A(k,k+N)=-r2/2;
B(k,k)=1-r1/2-r2/2;
B(k,k-1)=r1/2;
B(k,k+N)=r2/2;
%
for i=2:(N-1),
    A(i,i)=1+r1+r2/2;
    A(i,i-1)=-r1/2;
    A(i,i+1)=-r1/2;
    A(i,i+N)=-r2/2;
    B(i,i)=1-r1-r2/2;
    B(i,i-1)=r1/2;
    B(i,i+1)=r1/2;
    B(i,i+N)=r2/2;
end
% the middle layers
```

```

for j=2:(M-1),
    for i=2:(N-1),
        k=i+(j-1)*N;
        A(k,k)=1+r1+r2;
        A(k,k-1)=-r1/2;
        A(k,k+1)=-r1/2;
        A(k,k+N)=-r2/2;
        A(k,k-N)=-r2/2;
        B(k,k)=1-r1-r2;
        B(k,k-1)=r1/2;
        B(k,k+1)=r1/2;
        B(k,k+N)=r2/2;
        B(k,k-N)=r2/2;
    end
end
% left boundary
for j=2:(M-1),
    k=1+(j-1)*N;
    A(k,k)=1+r1/2+r2;
    A(k,k+1)=-r1/2;
    A(k,k+N)=-r2/2;
    A(k,k-N)=-r2/2;
    B(k,k)=1-r1/2-r2;
    B(k,k+1)=r1/2;
    B(k,k+N)=r2/2;
    B(k,k-N)=r2/2;
end
% the right boundary
for j=2:(M-1),
    k=N+(j-1)*N;
    A(k,k)=1+r1/2+r2;
    A(k,k-1)=-r1/2;
    A(k,k+N)=-r2/2;
    A(k,k-N)=-r2/2;
    B(k,k)=1-r1/2-r2;
    B(k,k-1)=r1/2;
    B(k,k+N)=r2/2;
    B(k,k-N)=r2/2;
end
% the upper boundary
k=1+N*(M-1);
A(k,k)=1+r1/2+r2/2;
A(k,k+1)=-r1/2;
A(k,k-N)=-r2/2;
B(k,k)=1-r1/2-r2/2;
B(k,k+1)=r1/2;
B(k,k-N)=r2/2;
%
k=N+N*(M-1);
A(k,k)=1+r1/2+r2/2;
A(k,k-1)=-r1/2;
A(k,k-N)=-r2/2;
B(k,k)=1-r1/2-r2/2;
B(k,k-1)=r1/2;
B(k,k-N)=r2/2;

```

```

%
for i=2:(N-1),
    k=i+N*(M-1);
    A(k,k)=1+r1+r2/2;
    A(k,k-1)=-r1/2;
    A(k,k+1)=-r1/2;
    A(k,k-N)=-r2/2;
    B(k,k)=1-r1-r2/2;
    B(k,k-1)=r1/2;
    B(k,k+1)=r1/2;
    B(k,k-N)=r2/2;
end
%% build the vector f
f=zeros(N*M,m);
f1=zeros(N,M);
for k=1:m,
    xc=0.5+0.25*cos(10*2*pi*t(k));
    yc=0.5+0.25*sin(10*2*pi*t(k));
    ff=a*exp(-(xx-xc).^2/(2*d^2)-(yy-yc).^2/(2*d^2))/(2*pi*d^2);
    f(:,k)=reshape(ff,N*M,1);
end
figure(2)
drawnow
for k=1:(m-1),
    b=B*u+dt*(f(:,k)+f(:,k+1))/2;
    u=A\b;
    pause(0.2)
    temp=reshape(u,M,N);
    h=surf(xx,yy,temp)
    set(h,'edgecolor','none','facecolor','interp');
    %view(2)
    drawnow
end

```

THIS PAGE INTENTIONALLY LEFT BLANK

## APPENDIX E. FFT CODE FOR 2-D SIMULATION

```

clear
%
N=128; % number of numerical points in [0, 1]
dx=1/N; % spatial step
x=[0:N]'*dx; % numerical grid
[xa, ya]=meshgrid(x,x);
u=zeros(N+1,N+1); % solution at current time, u(j)=u(x(j))
d=0.02; % radius of Gaussian source
a=1; % magnitude of Gaussian source
u0=zeros(N+1,N+1); % initial value
u=u0;
%
dt=dx/8; % time step
T=12/128;
m=T/dt; % number of time steps
t=[0:m]*dt;
%
% going to the coefficients of cosin expansion
w=[u; u(N:-1:2,:)];
w=[w,w(:,N:-1:2)];
z=fft2(w);
cu=real(z(1:N+1,1:N+1))/N^2;
Na=[0:N];
[Nx, Ny]=meshgrid(Na, Na);
r=(Nx*pi).^2+(Ny*pi).^2;
r(1,1)=1;
h1=(1-exp(-r*dt))./r;
h1(1,1)=dt;
h2=(1-(r*dt+1).*exp(-r*dt))./r.^2;
h2(1,1)=0.5*dt^2;
r(1,1)=0;
%
x0=0.5+0.25*cos(10*2*pi*0);
y0=0.5+0.25*sin(10*2*pi*0); % location of Gaussian source at t^{k-1}
f0=a*exp(-((xa-x0).^2+(ya-y0).^2)/(2*d^2))/(2*pi*d^2); % Gaussian
source at t^{k-1}
w=[f0; f0(N:-1:2,:)];
w=[w,w(:,N:-1:2)];
z=fft2(w);
cf0=real(z(1:N+1,1:N+1))/N^2;
%
surf(xa, ya, u, 'edgecolor','none','facecolor','interp')
axis([0, 1, 0, 1, -0.04,0.4])
caxis([0, 0.4])
view(3)
drawnow
for k=1:m,
    x1=0.5+0.25*cos(10*2*pi*t(k+1));
    y1=0.5+0.25*sin(10*2*pi*t(k+1)); % location of Gaussian source at t^k
    f1=a*exp(-((xa-x1).^2+(ya-y1).^2)/(2*d^2))/(2*pi*d^2); % Gaussian
    source at t^k

```

```

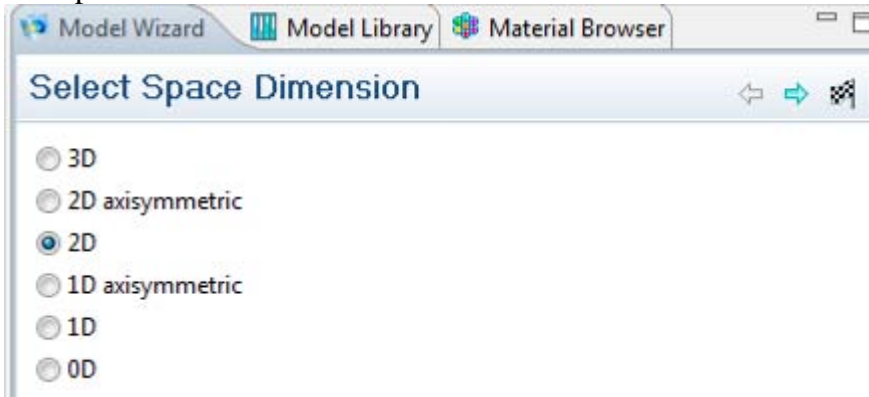
w=[f1; f1(N:-1:2,:)];
w=[w,w(:,N:-1:2)];
z=fft2(w);
cf1=real(z(1:N+1,1:N+1))/N^2;
% update cu
cu=cu.*exp(-r*dt)+cf1.*h1+(cf0-cf1)/dt.*h2;
cf0=cf1;
k
% going back to the function
z=N^2*[cu; cu(N:-1:2,:)];
z=[z,z(:,N:-1:2)];
w=ifft2(z);
u=real(w(1:N+1,1:N+1));
%pause(0.2)
surf(xa, ya, u, 'edgecolor','none','facecolor','interp')
axis([0, 1, 0, 1, -0.04,0.4])
caxis([0, 0.4])
view(2)
drawnow
end

```

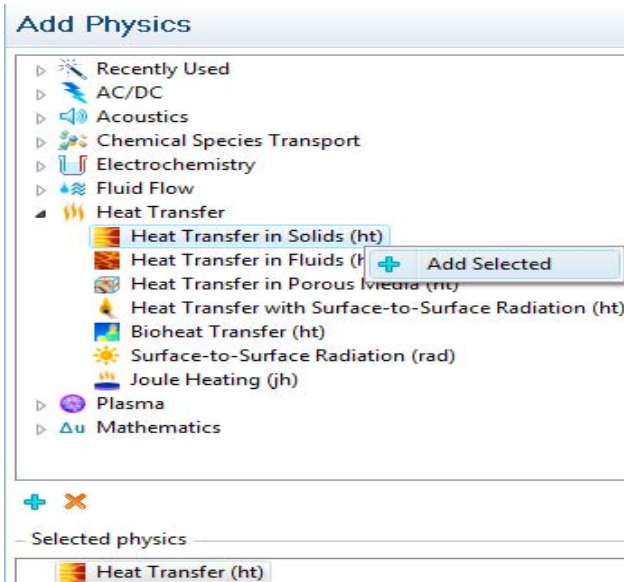


## APPENDIX F. COMSOL CODE FOR 2-D SIMULATION

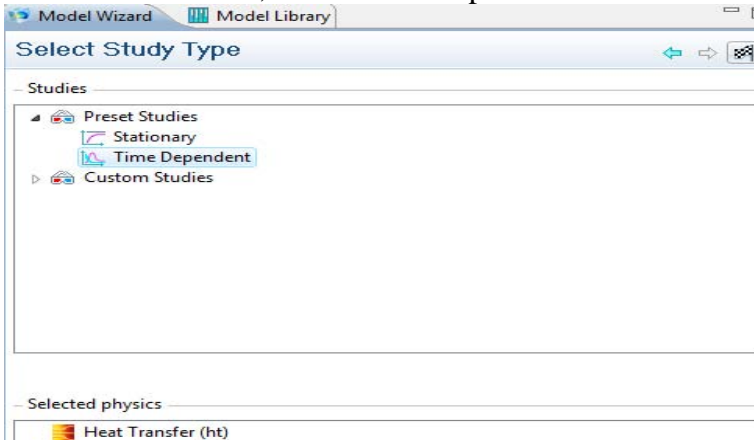
2. Open COMSOL 4.0a with 2D and hit ➡



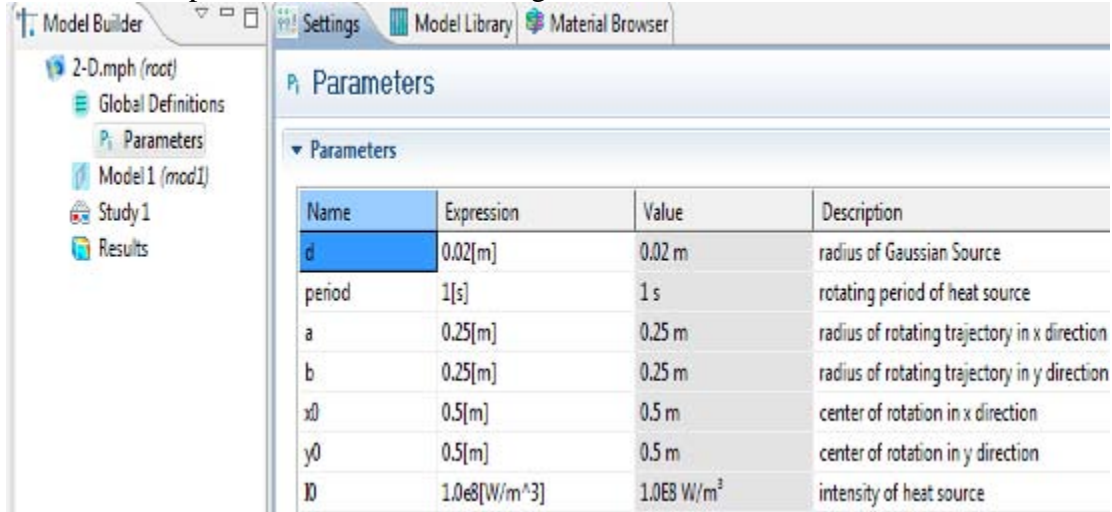
2. In Heat Transfer, select Heat Transfer in Solids (ht) and right-click Add selected then hit ➡.



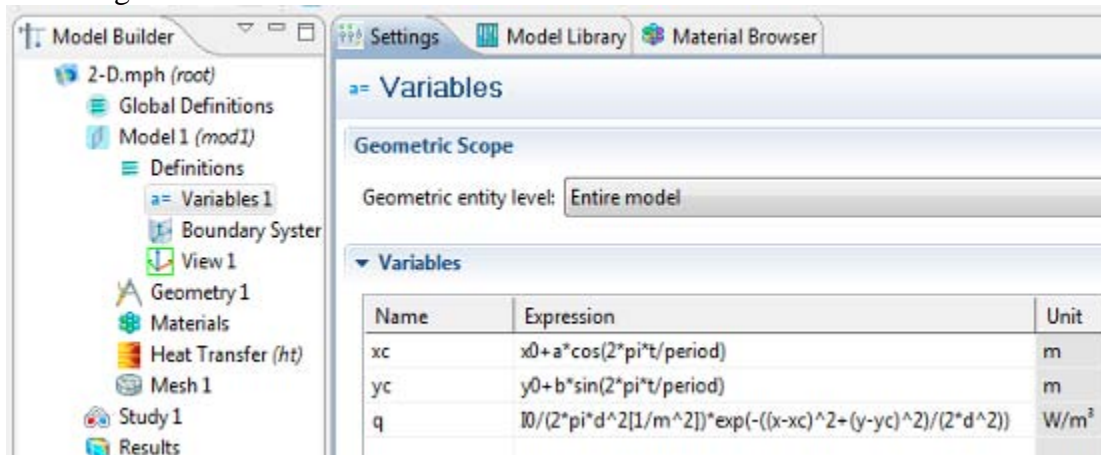
3. In Preset Studies, select Time Dependent and hit Finish 🏁.




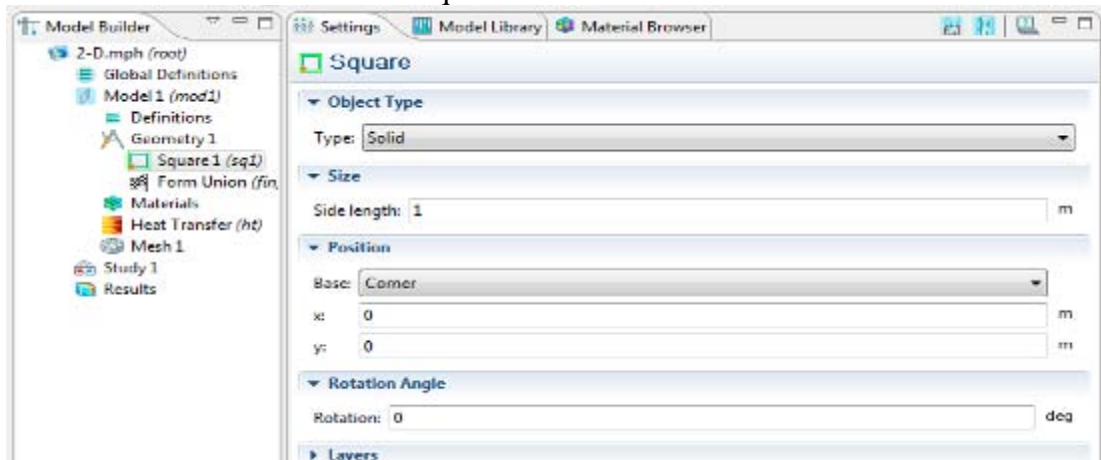
4. Under Model Builder, right-click on Global Definitions and left-click to select Parameters, input Parameters as following:



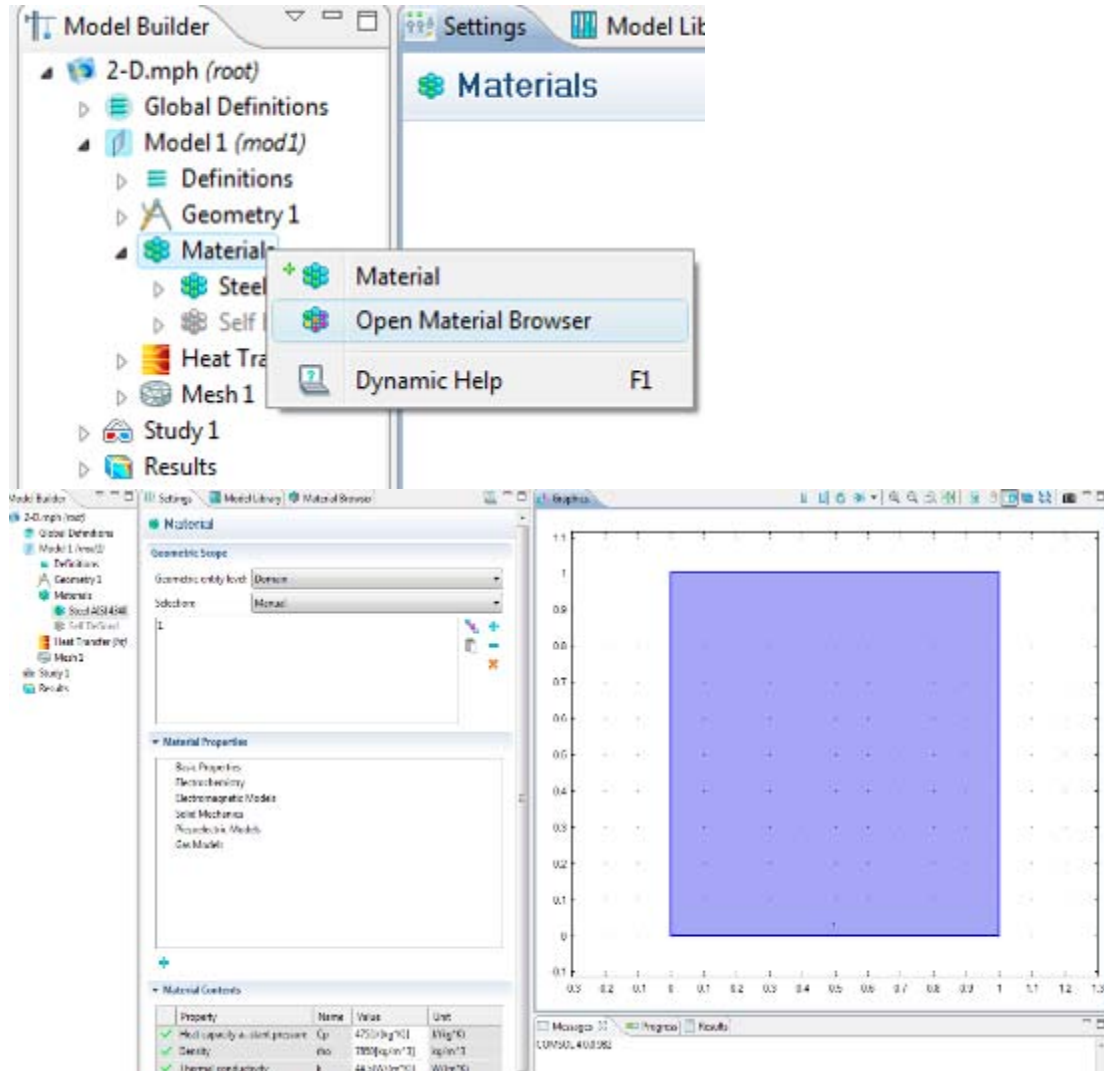
5. Right-click on Definitions and left-click to select Variables input Variables as following:



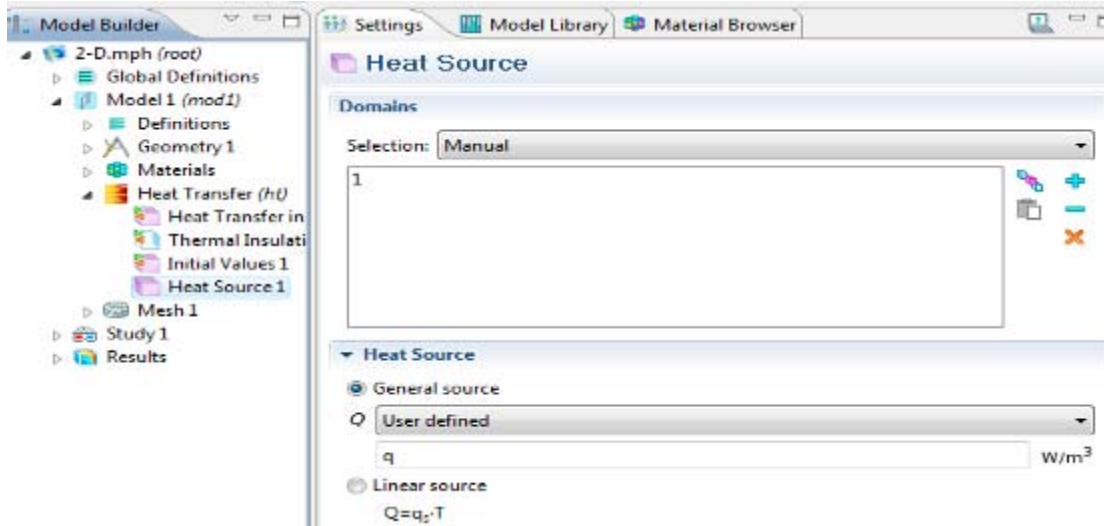
6. Right-click on Geometry and left-click square and make side length 1 m. Click build all button  to create a square.



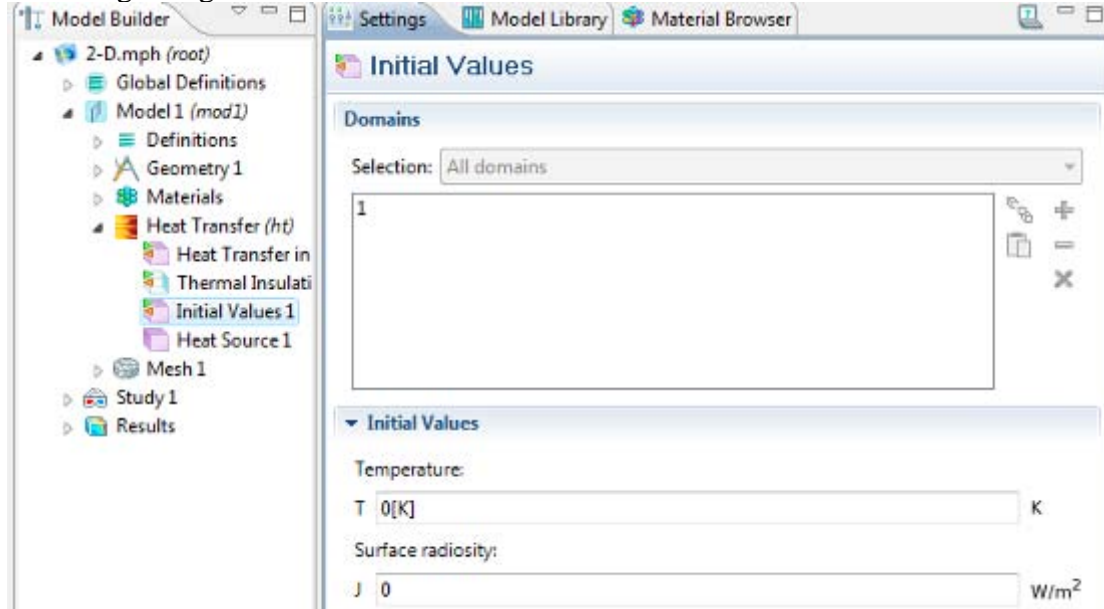
7. Right-click on Materials and left-click on add Materials, add Steel AISI 4340. Right-click on the “square” in Graphics and left-click on it to select. Make sure that “1” is under the selection.




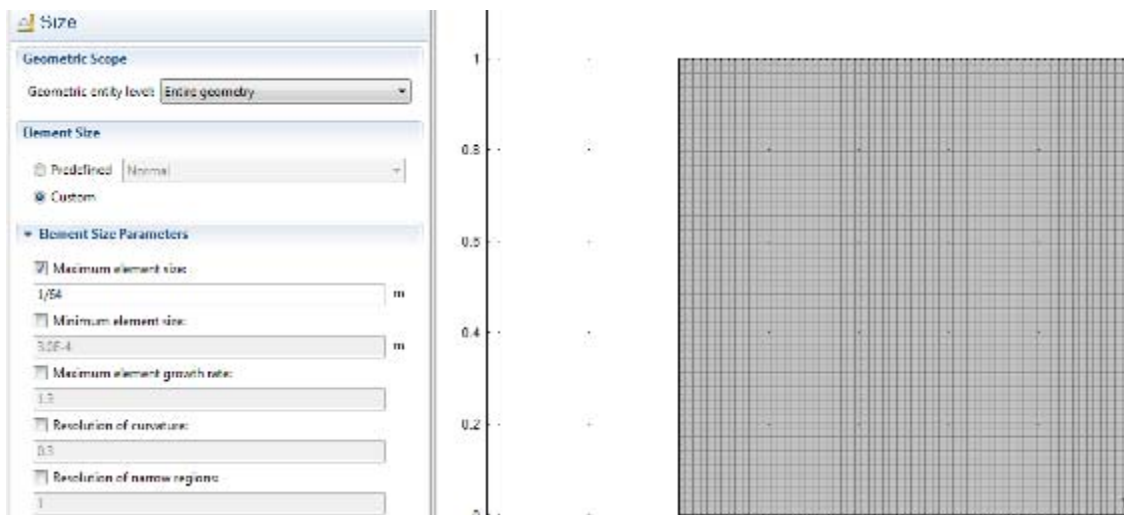
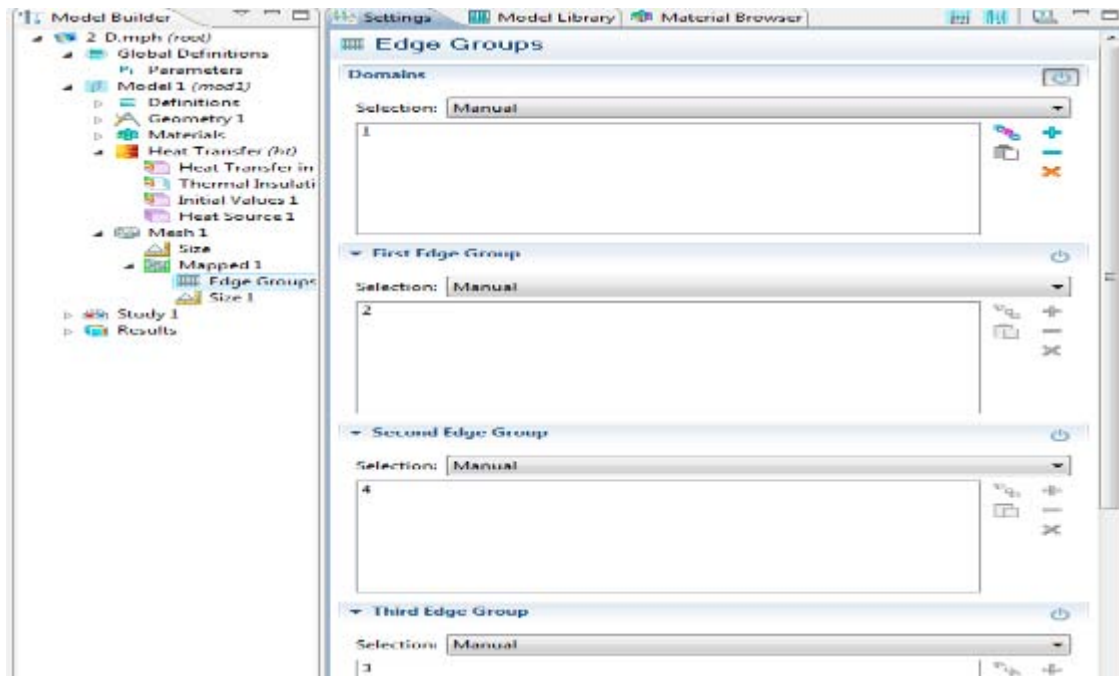
8. Right-click on Heat Transfer and left-click to add Heat Source. For General source Q, Select User defined and Put “q”, which is the heat source defined from the Variable.




9. In Initials Values, make Temperature equal 0. We assume there is no temperature rise in the beginning.

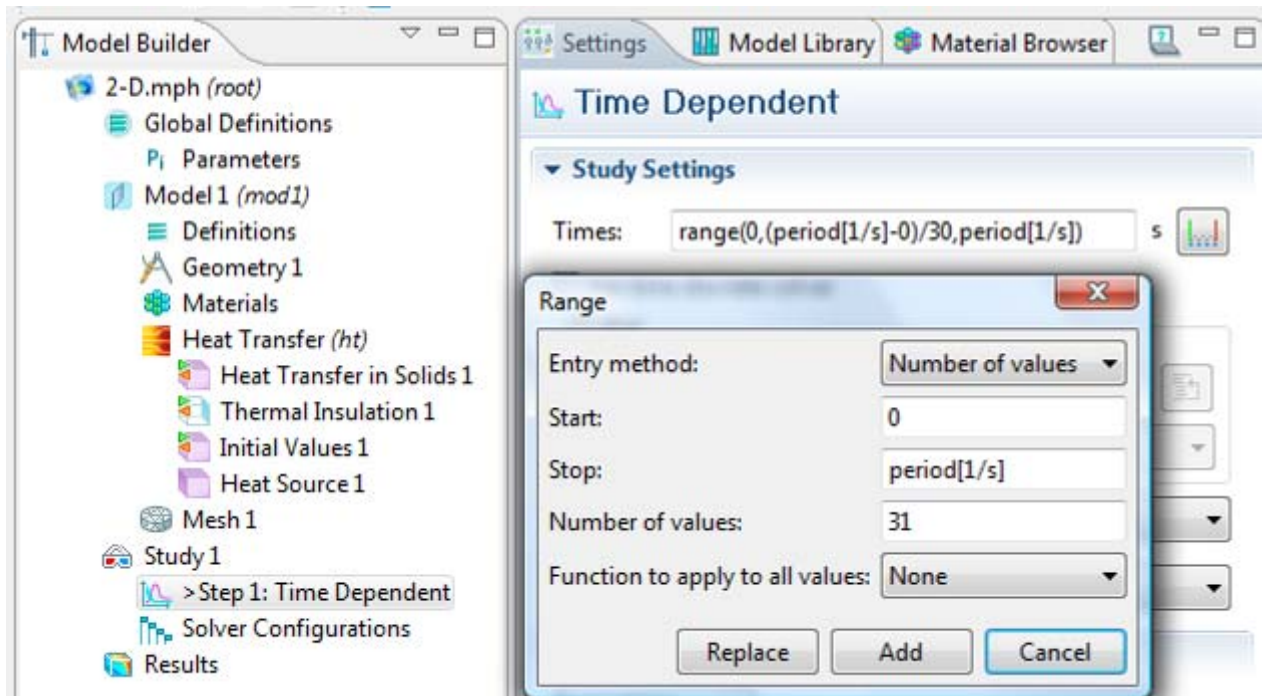



10. Right-click on Mesh and select Mapped. Right-click on Mapped and select Edge Groups. Please select domain and add each edge group. Under size, select Custom and make Maximum element size equal  $1/64$  or any  $1/2^N$  where  $N$  is an integer to compare with FFT 2 method in MATLAB. Then select build all button  to build the mesh.

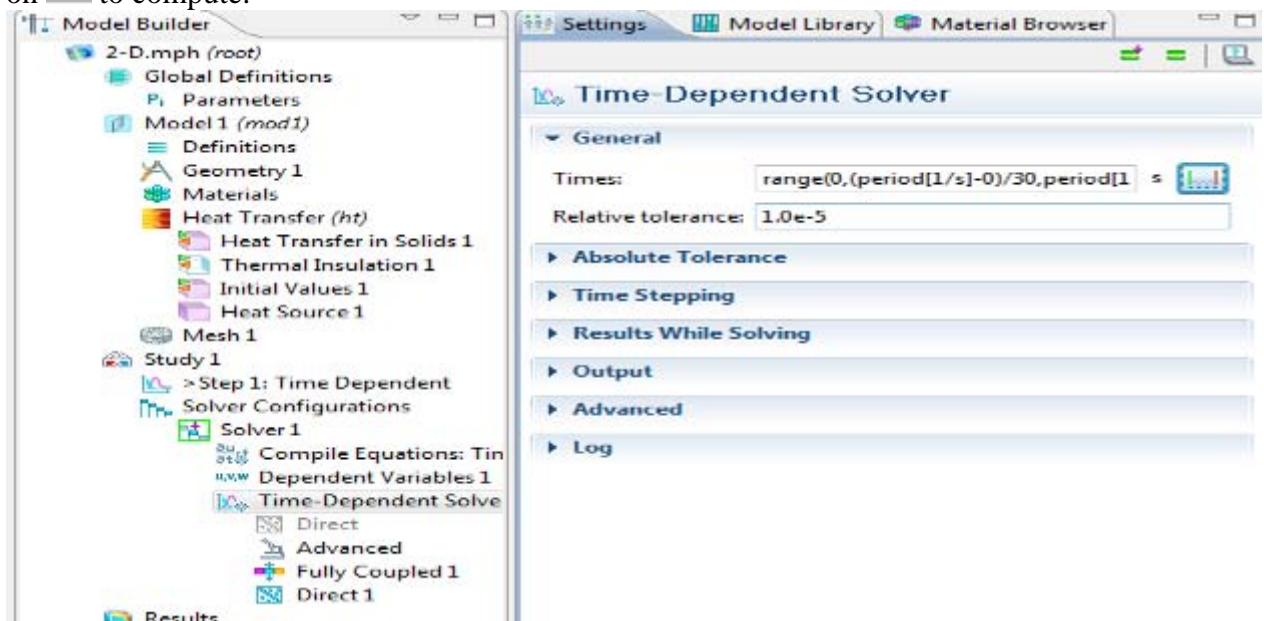


11. Under Study in Step, select Range button  . Under Entry method, select Number of values, start from 0 and stop at “period” which is defined in the parameters. Put Number of values to be 31.

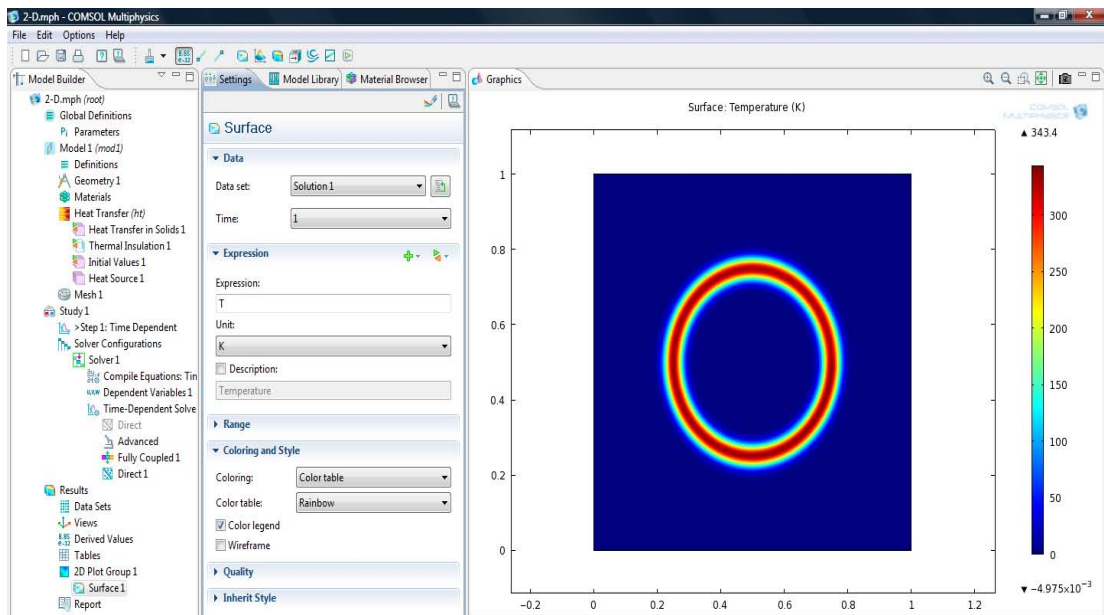




12. Under Time-Dependent Solver, make Relative tolerance to be  $1.0e-5$ . Left-click on  to compute.



14. Finally, under Results, select Surface under 2D Plot Group and select time to be 1 only, we should be able to see the result like Figure 25.



14. Further data analysis can be done under Report, such as generate a movie and export data and further compare results with MATLAB.

THIS PAGE INTENTIONALLY LEFT BLANK

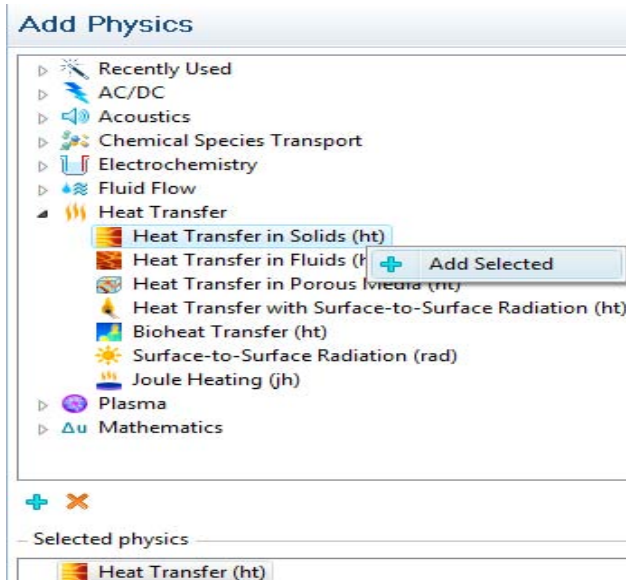


## APPENDIX G. COMSOL CODE FOR 3-D SIMULATION

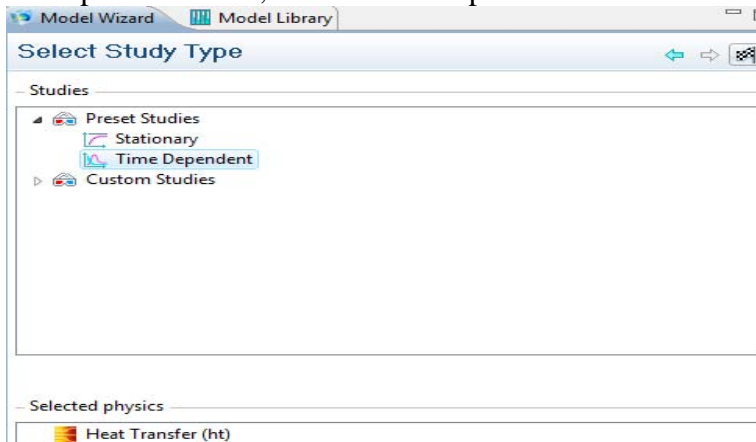
1. Open COMSOL 4.0a with 3D and hit ➡



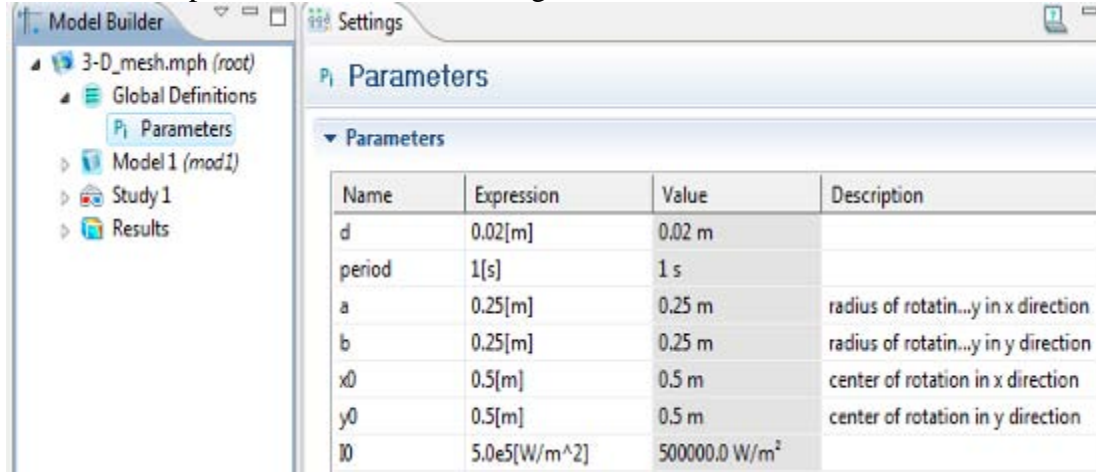
2. In Heat Transfer, select Heat Transfer in Solids (ht) and right-click Add selected then hit ➡.



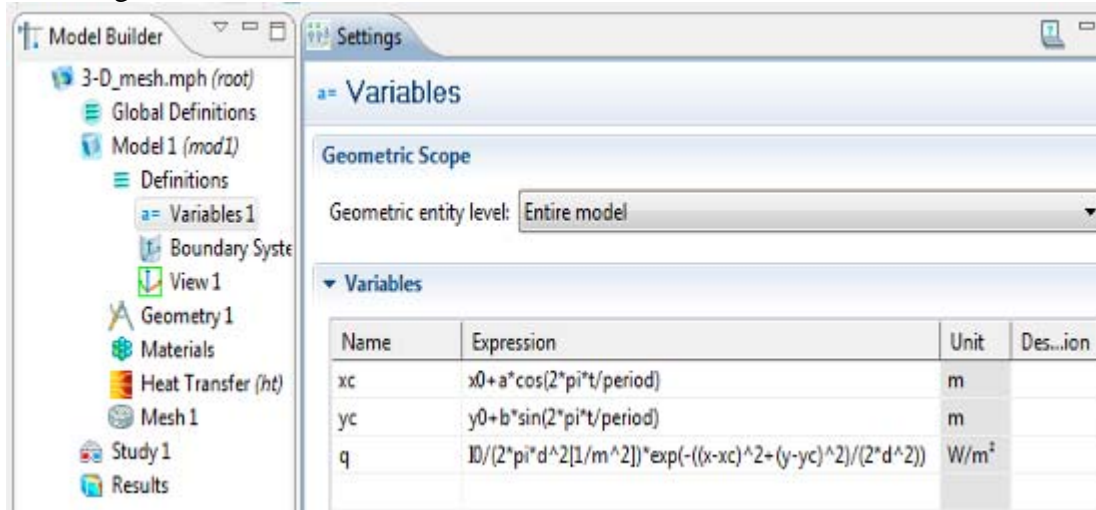
3. In preset Studies, select Time Dependent and hit Finish ⏏.



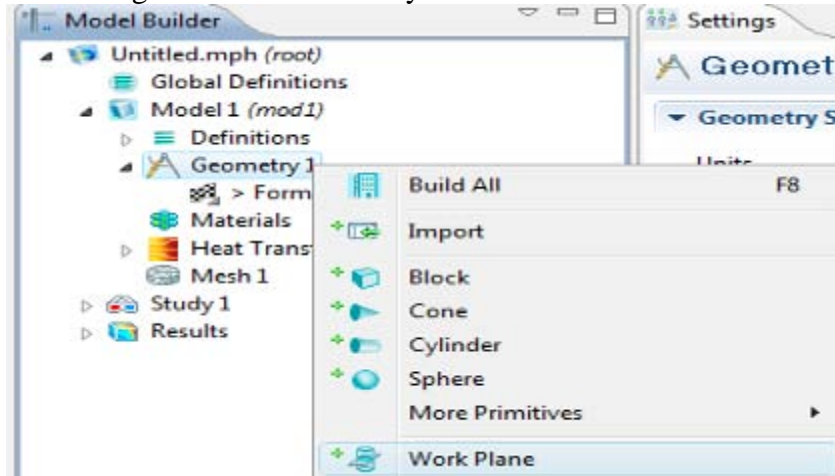
4. Under Model Builder, right-click on Global Definitions and left-click to select Parameters, input Parameters as following:



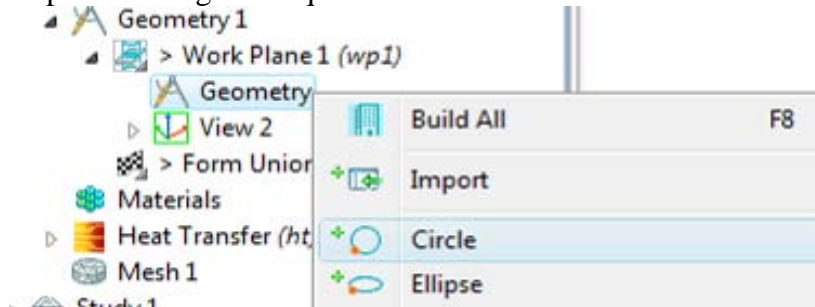
5. Right-click on Definitions and left-click to select Variables. Input Variables as following:



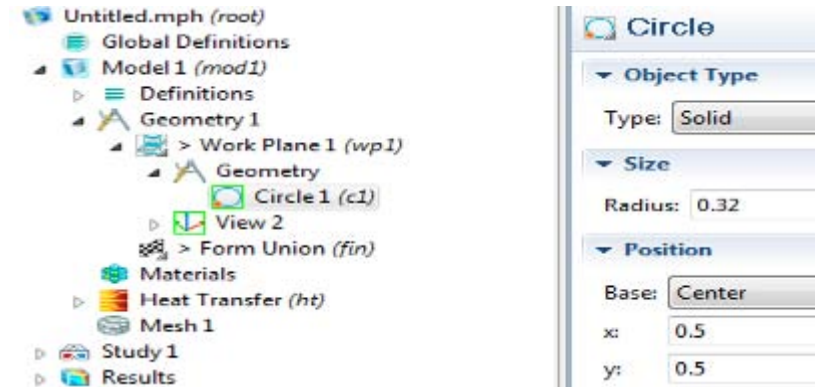
6. Right-click on Geometry and left-click “Work Plane” and input 1 in z coordinate.



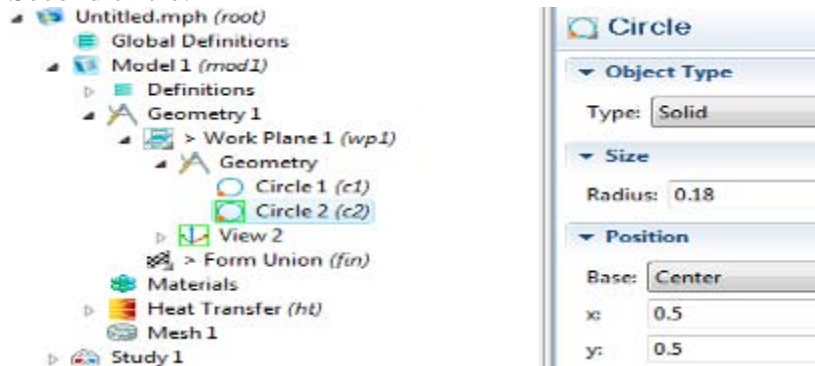
7. Right-click on Geometry under Work Plane, select Circle. We need to build up two circles. One has radius 0.32m and the other one has radius 0.18m, both are centered at  $x=0.5\text{m}$  and  $y=0.5\text{m}$ . Right-click on Geometry and add Extrude, select Reverse direction and input distance to be 0.05m. Right-click on Geometry and add Block with Width, Depth and Height all equal 1.



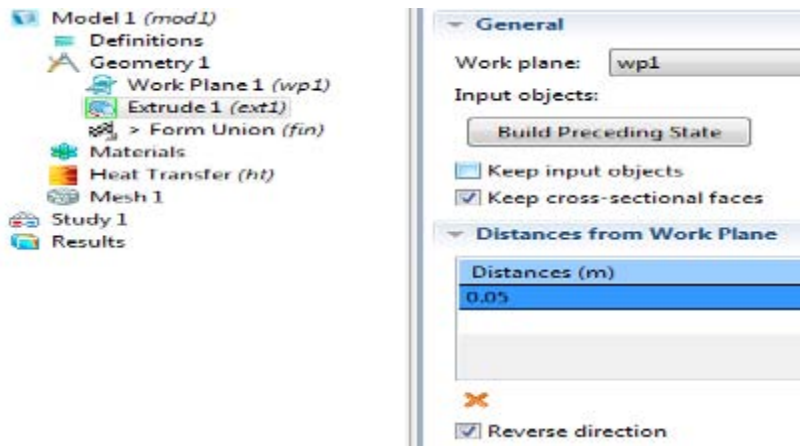
First circle:



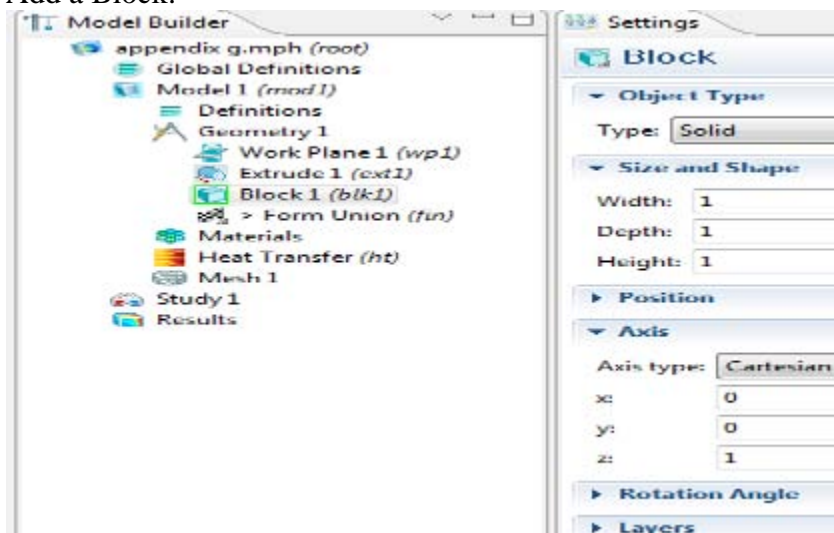
Second circle:



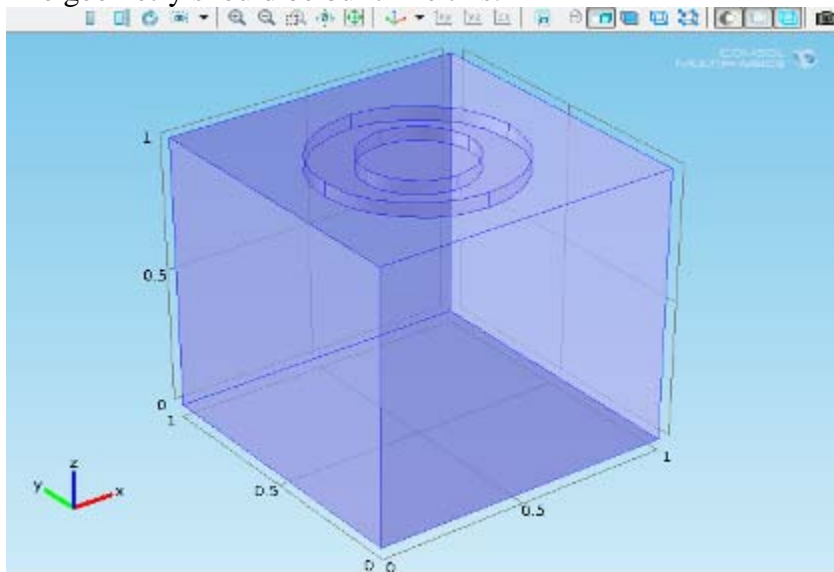
Extrude:



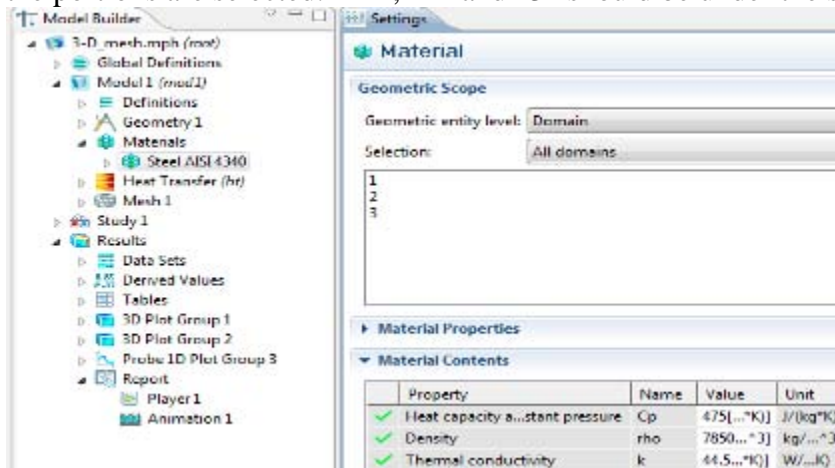
Add a Block:



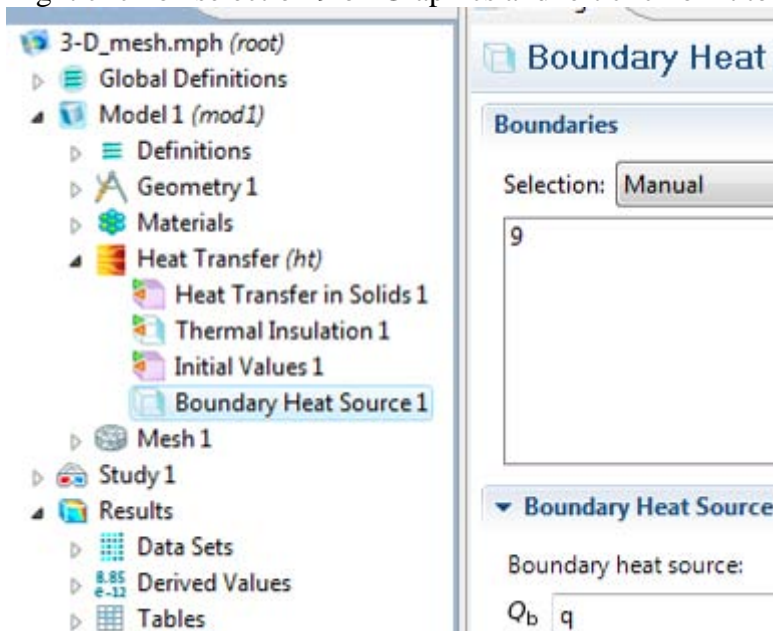
The geometry should be built like this:



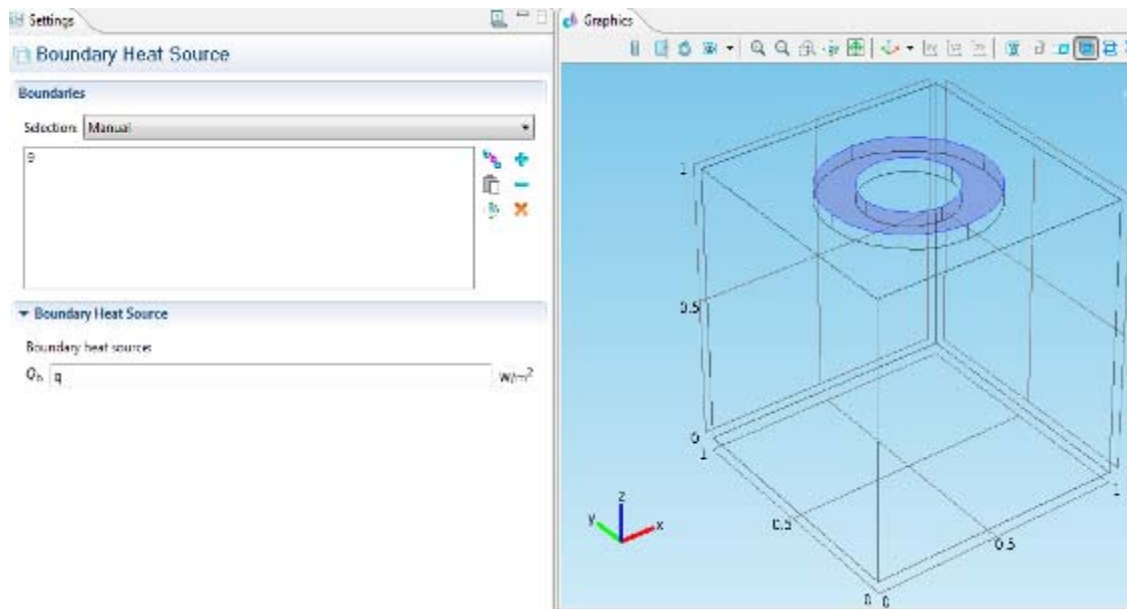
7. Right-click on Materials and left-click on add Materials. Add Steel AISI 4340. Right-click on the geometry in Graphics and left-click on it to select. Make sure that all the portions are selected. “1”, “2” and “3” should be under the selection.



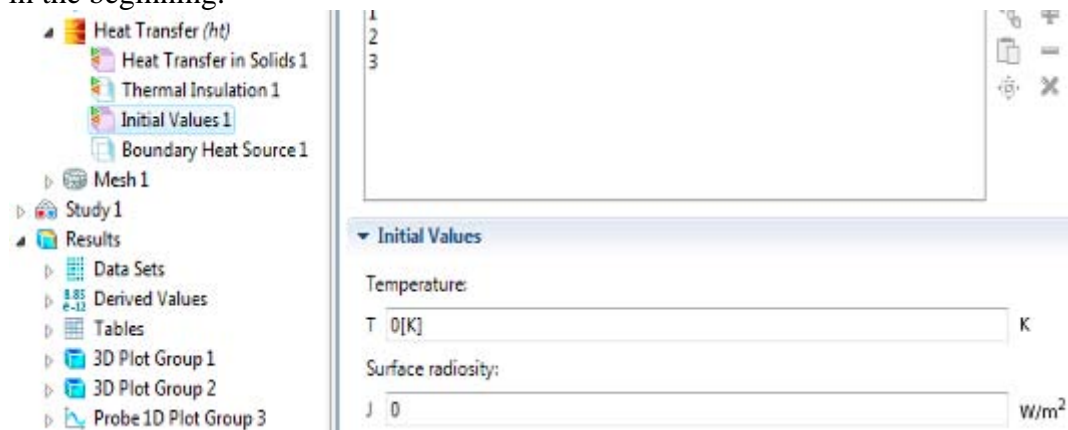
8. Right-click on Heat Transfer and left-click to add Boundary Heat Source. For Boundary heat source  $Q_b$ , enter “q”, which is the heat source defined from the Variable. Right click on selection 9 on Graphics and left click on it to select.



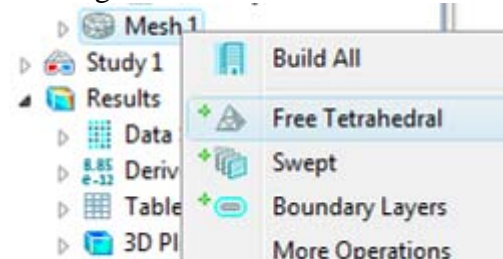




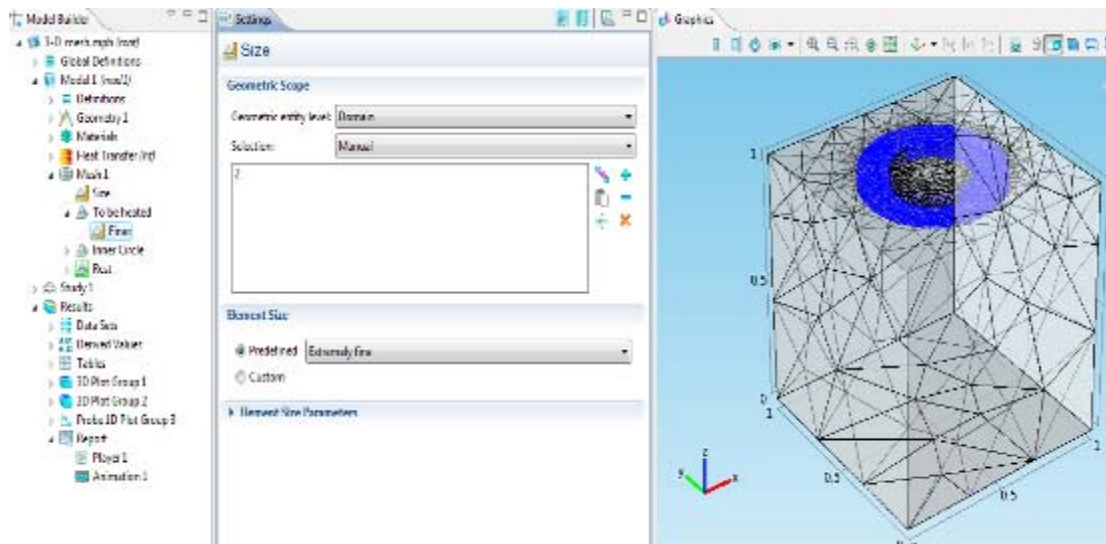
9. In Initials Values, make Temperature equal 0. We assume there is no temperature rise in the beginning.



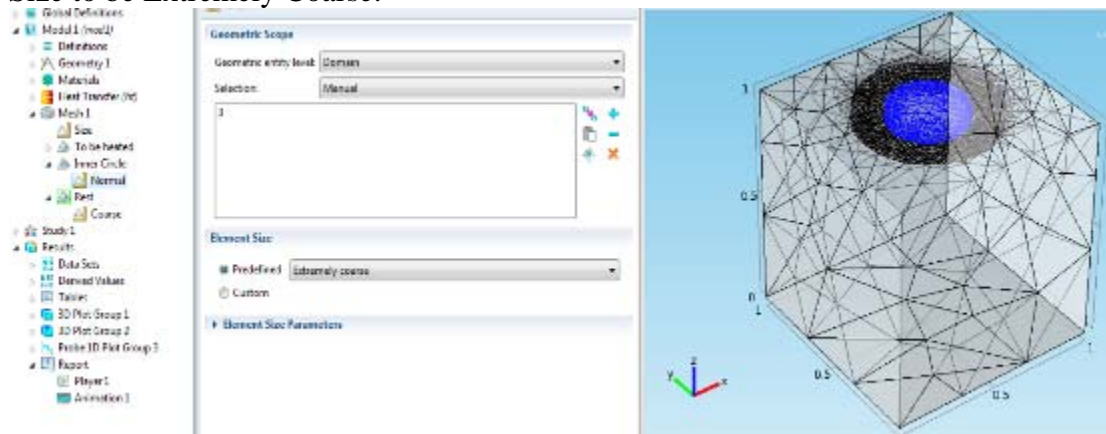
10. Right-click on Mesh and select Free Tetrahedral, repeat this process three times.



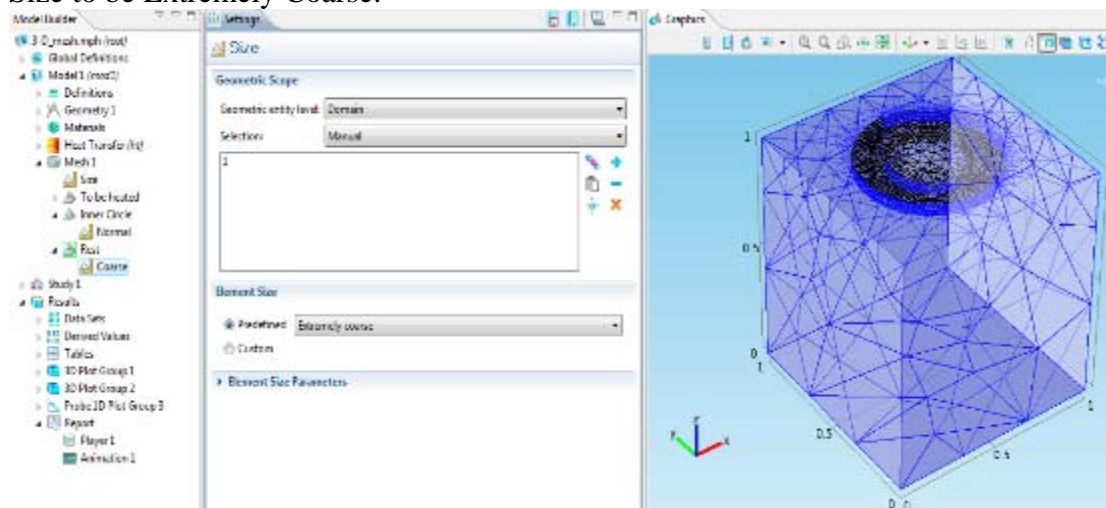
First Free Tetrahedral mesh: selection 2 is picked and makes the predefined Element Size to be Extremely Fine:




Second Free Tetrahedral mesh: selection 3 is picked and makes the predefined Element Size to be Extremely Coarse:

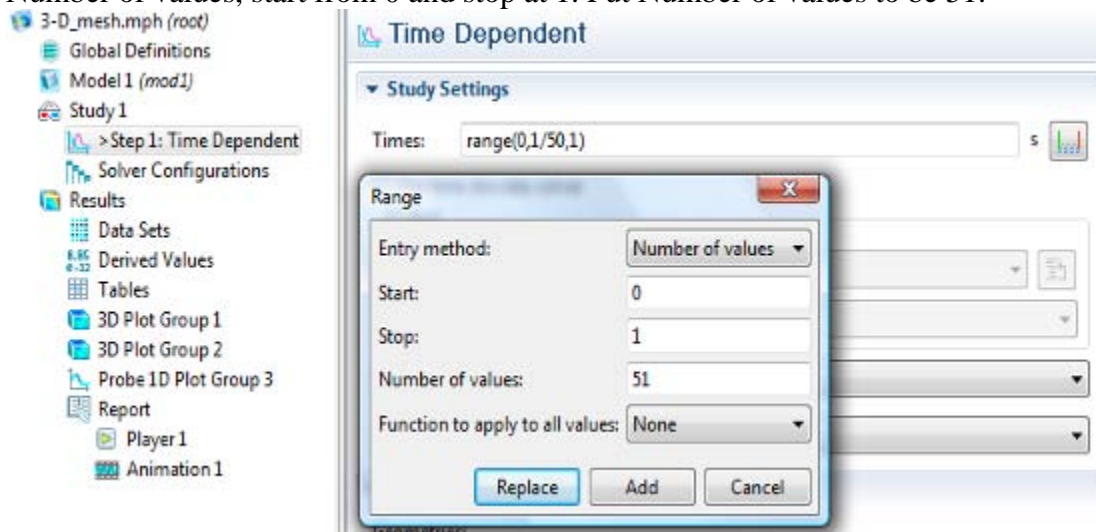



Third Free Tetrahedral mesh: selection 1 is picked and makes the predefined Element Size to be Extremely Coarse.

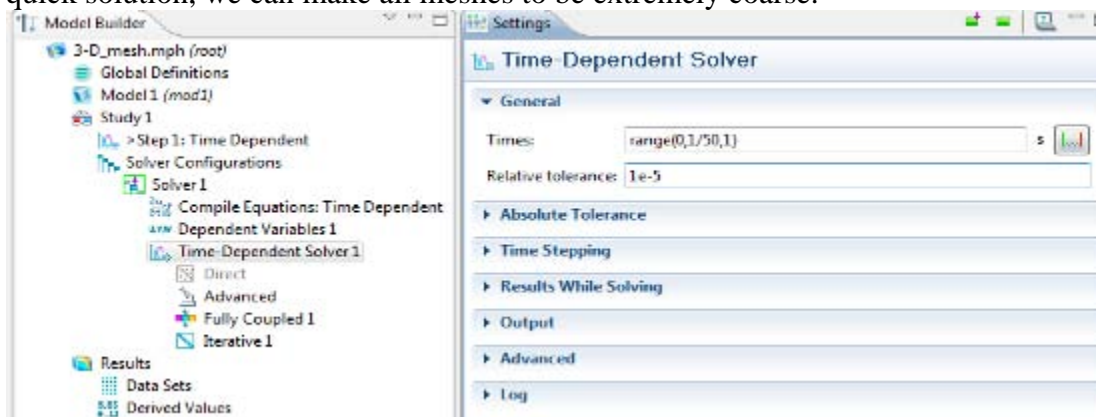


Then select build all button  to build the mesh.

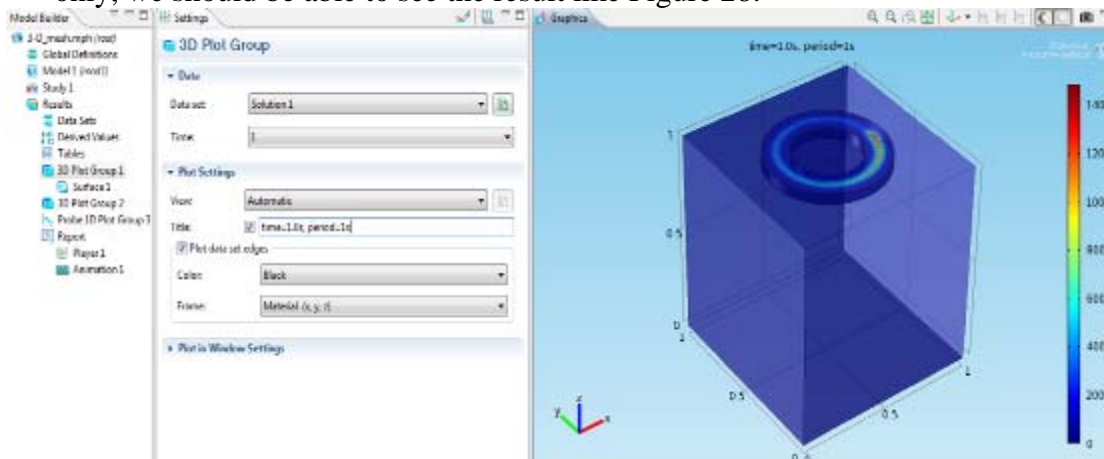
11. Under Study in Step, select Range button , under Entry method, select Number of values, start from 0 and stop at 1. Put Number of values to be 51.



12. Under Time-Dependent Solver, make Relative tolerance to be  $1.0e-5$ . Left-click on  to compute. This 3D problem may take couple hours to solve, in order to see a quick solution, we can make all meshes to be extremely coarse.

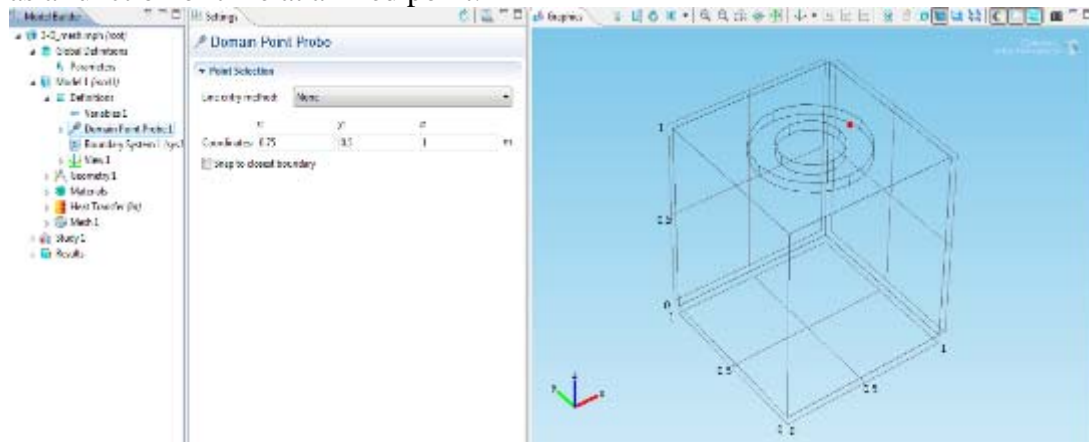


2. Finally, under Results, select Surface under 3D Plot Group and select time to be 1 only, we should be able to see the result like Figure 28.

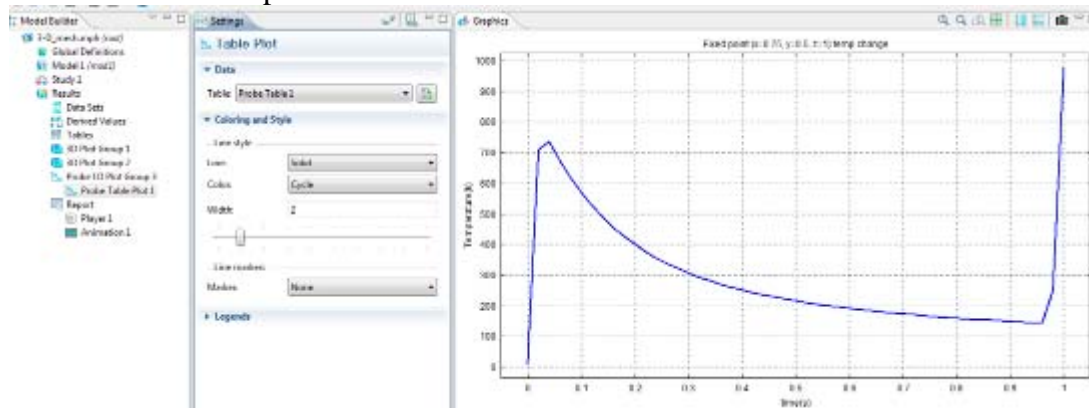




14. Further data analysis can be done under Report, such as generate a movie and export data. We can add a Domain Probe Point under Definition to analyze the temperature rise as a function of time at a fixed point:



Click on Probe 1D Plot Group under Results, we can plot temperature as a function of time at the selected point.



THIS PAGE INTENTIONALLY LEFT BLANK

## LIST OF REFERENCES

- [1] G. Araya, and G. Gutierrez, “Analytical solution for a transient, three-dimensional temperature distribution due to a moving laser beam,” *International Journal of Heat and Mass Transfer*, vol. 49, pp. 4124–4131, June 2006.
- [2] J. E. Moody and R. H. Hendel, “Temperature profiles induced by a scanning cw laser beam,” *Journal of Applied Physics*, vol. 53(6), pp. 4364–4371, June 1982.
- [3] M. Lax. “Temperature rise induced by a laser beam II. The nonlinear case,” *Applied Physics, Letters*, vol. 33(8), pp. 786–788, Oct. 1978.
- [4] M. L. Burgener and R. E. Reedy, “Temperature distributions produced in a two-layer structure by a scanning cw laser or electron beam.” *Journal of Applied Physics*, vol. 53(6), pp. 4357–4363, June 1982.
- [5] H. Zhou, “Temperature rise induced by a rotating or dithering laser beam,” Preprint, Department of Applied Mathematics, Naval Postgraduate School, Oct. 2010.
- [6] W. Silfvast, *Laser Fundamentals*, Second Edition. Cambridge University Press, 2008.
- [7] R. Haberman, *Applied Partial Differential Equations with Fourier Series and Boundary Value Problems*, Fourth Edition, Person Prentice Hall, 2004.
- [8] A. Larraza, “Lasers, Optoelectronics and Electro-Optics I,” PowerPoint presentation slides, Department of Physics, Naval Postgraduate School, 2010.
- [9] N. Bianco, O. Manca, S. Nrdini, and S. Tamburrino, “A numerical model for transient heat conduction in semi-infinite solids irradiated by a moving heat source,” Excerpts from the Proceedings of the COMSOL Conference, Hannover, 2008.

THIS PAGE INTENTIONALLY LEFT BLANK

## INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center  
Ft. Belvoir, Virginia
2. Dudley Knox Library  
Naval Postgraduate School  
Monterey, California
3. Professor Hong Zhou  
Department of Applied Mathematics  
Naval Postgraduate School  
Monterey, California
4. Professor Gamani Karunasiri  
Department of Physics  
Naval Postgraduate School  
Monterey, California
5. Professor Carlos Borges  
Department of Applied Mathematics  
Naval Postgraduate School  
Monterey, California
6. Professor Andres Larraza  
Department of Physics  
Naval Postgraduate School  
Monterey, California